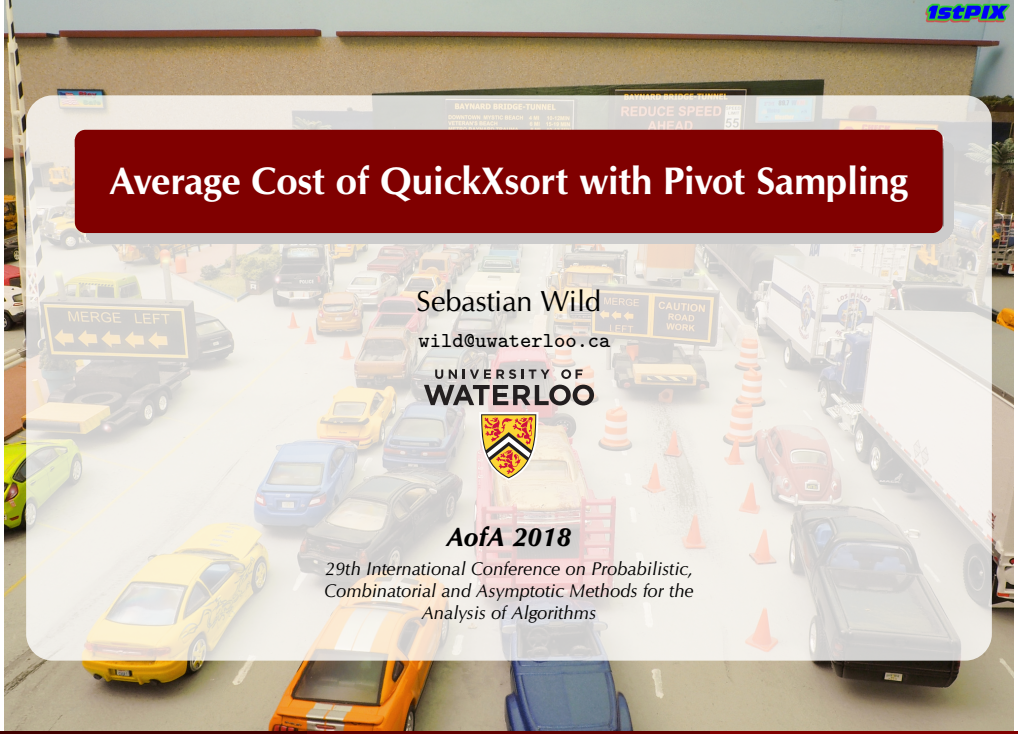


Average Cost of QuickXsort with Pivot Sampling

Sebastian Wild

wild@uwaterloo.ca

UNIVERSITY OF WATERLOO

AofA 2018

29th International Conference on Probabilistic, Combinatorial and Asymptotic Methods for the Analysis of Algorithms

Outline

1 What is QuickXsort?

2 Recurrence

3 Toll function

4 Result

What is QuickXsort?

$$\text{Quick}\mathbf{X}\text{sort} = \text{Quicksort} + \mathbf{X}$$

What is QuickXsort?

QuickXsort = Quicksort + X

QuickXsort(A[l..r])

1. Choose pivot P.
2. i = Partition(A[l..r], P)
   // subproblems: A[l..i − 1] and A[i + 1..r]
3. Sort one subproblem with X.
4. Sort the other subproblem with QuickXsort.

What is QuickXsort?

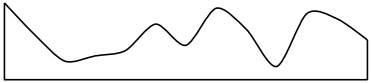
QuickXsort = Quicksort + X

QUICKXSORT($A[l..r]$)

1. Choose pivot P.
2. i = PARTITION($A[l..r]$, P)
   // subproblems: $A[l..i-1]$ and $A[i+1..r]$
3. Sort one subproblem with X.
4. Sort the other subproblem with QUICKXSORT.

What is QuickXsort?

QuickXsort = Quicksort + **X**



QUICKXSORT($A[l..r]$)

1 Choose pivot P. ← median of $k = 2t + 1$

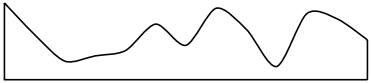
2 i = PARTITION($A[l..r]$, P)
// subproblems: $A[l..i-1]$ and $A[i+1..r]$

3 Sort one subproblem with X.

4 Sort the other subproblem with QUICKXSORT.

What is QuickXsort?

QuickXsort = Quicksort + **X**



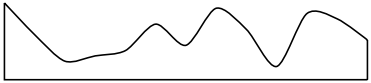
QUICKXSORT($A[l..r]$)

median of $k = 2t + 1$

1. Choose pivot P.
2. i = PARTITION($A[l..r]$, P)
   // subproblems: $A[l..i-1]$ and $A[i+1..r]$
3. Sort one subproblem with X.
4. Sort the other subproblem with QUICKXSORT.

What is QuickXsort?

QuickXsort = Quicksort + **X**



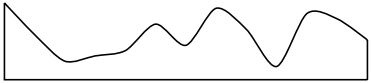
QUICKXSORT($A[l..r]$)

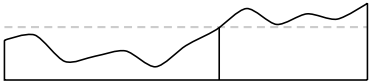
median of $k = 2t + 1$

1. Choose pivot P.
2. i = PARTITION($A[l..r]$, P)
   // subproblems: $A[l..i-1]$ and $A[i+1..r]$
3. Sort one subproblem with X.
4. Sort the other subproblem with QUICKXSORT.

What is QuickXsort?

QuickXsort = Quicksort + **X**





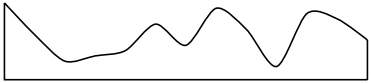
QUICKXSORT($A[l..r]$)

median of $k = 2t + 1$

1. Choose pivot P.
2. i = PARTITION($A[l..r]$, P)
   // subproblems: $A[l..i-1]$ and $A[i+1..r]$
3. Sort one subproblem with X.
4. Sort the other subproblem with QUICKXSORT.

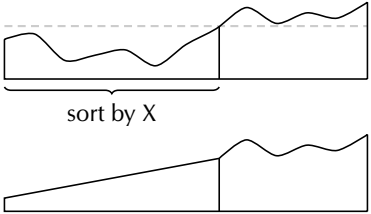
What is QuickXsort?

QuickXsort = Quicksort + **X**



QUICKXSORT($A[l..r]$)

1. Choose pivot P. ← median of $k = 2t + 1$
2. i = PARTITION($A[l..r]$, P)
   // subproblems: $A[l..i-1]$ and $A[i+1..r]$
3. **Sort one subproblem with X.**
4. Sort the other subproblem with QUICKXSORT.

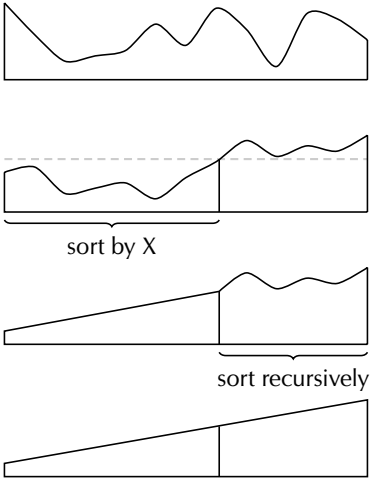


What is QuickXsort?

QuickXsort = Quicksort + **X**

QUICKXSORT($A[l..r]$)

1. Choose pivot P. ← median of $k = 2t + 1$
2. i = PARTITION($A[l..r]$, P)
   // subproblems: $A[l..i-1]$ and $A[i+1..r]$
3. **Sort one subproblem with X.** ← Which one?
4. Sort the other subproblem with QUICKXSORT.



What is QuickXsort?

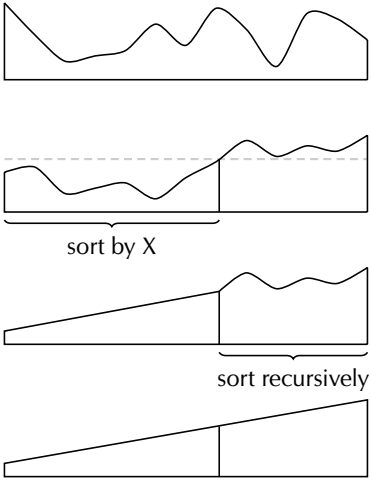
QuickXsort = Quicksort + **X**

QUICKXSORT($A[l..r]$)

1. Choose pivot P. ← median of $k = 2t + 1$
2. i = PARTITION($A[l..r]$, P)
   // subproblems: $A[l..i-1]$ and $A[i+1..r]$
3. **Sort one subproblem with X.**
4. Sort the other subproblem with QUICKXSORT.

Which one?

And why should that help?!



What is QuickXsort?

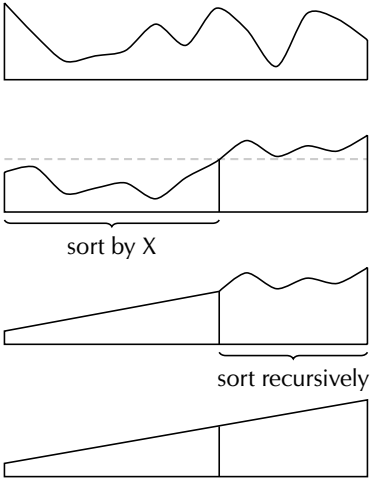
QuickXsort = Quicksort + **X**

QUICKXSORT($A[l..r]$)

1. Choose pivot P. ← median of $k = 2t + 1$
2. i = PARTITION($A[l..r]$, P)
   // subproblems: $A[l..i-1]$ and $A[i+1..r]$
3. **Sort one subproblem with X,**
   using the other subproblem as scratch space.
4. Sort the other subproblem with QUICKXSORT.

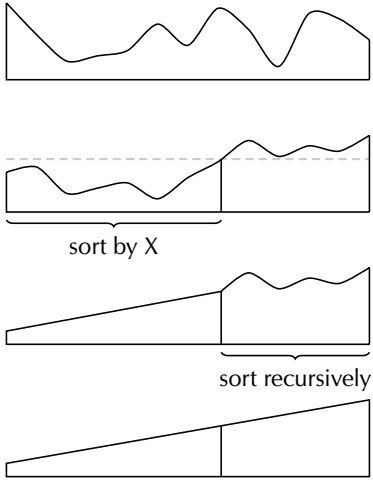
Which one?

And why should that help?!



What is QuickXsort?

QuickXsort = Quicksort + **X**



QUICKXSORT($A[l..r]$)

1. Choose pivot P. — median of $k = 2t + 1$
2. i = PARTITION($A[l..r]$, P)
   // subproblems: $A[l..i-1]$ and $A[i+1..r]$
3. **Sort one subproblem with X,**
   using the other subproblem as scratch space.
4. Sort the other subproblem with QUICKXSORT.

Which one?

And why should that help?!

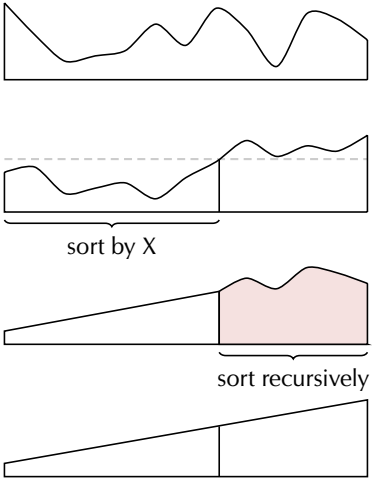
⇝ *QuickXsort can turn X into an in-place sorting method!*

What is QuickXsort?

QuickXsort = Quicksort + **X**

QUICKXSORT($A[l..r]$)

1. Choose pivot P. ← median of $k = 2t + 1$
2. i = PARTITION($A[l..r]$, P)
   // subproblems: $A[l..i-1]$ and $A[i+1..r]$
3. **Sort one subproblem with X,**
   using the other subproblem as scratch space.
4. Sort the other subproblem with QUICKXSORT.



⇝ *QuickXsort can turn X into an in-place sorting method!*

QuickMergesort

Good candidate for X: ***Mergesort***

- #comparisons almost optimal
- but: merging uses extra space as buffer
- merging possible without extra space
  - but not really practical!

"Worst merge lanes in the D.C. area"

QuickMergesort

Good candidate for X: ***Mergesort***

- #comparisons almost optimal
- but: merging uses extra space as buffer
- merging possible without extra space
  - but not really practical!

"Worst merge lanes in the D.C. area"

QuickMergesort

Good candidate for X: ***Mergesort***

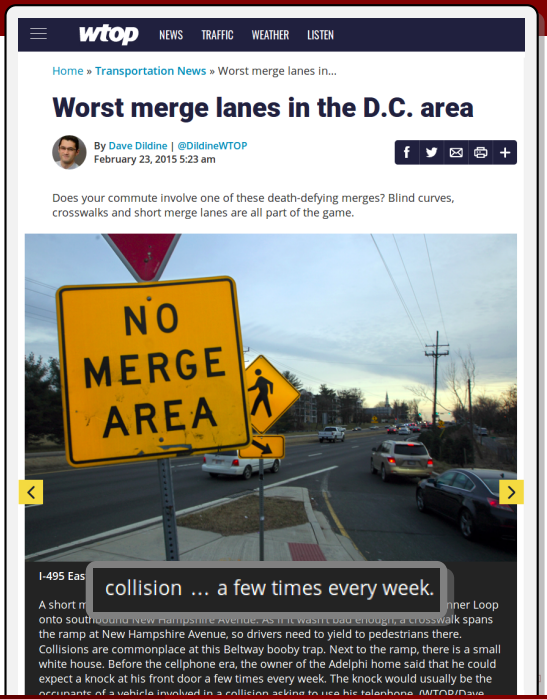
- #comparisons almost optimal
- but: merging uses extra space as buffer
- merging possible without extra space
  - but not really practical!

"Worst merge lanes in the D.C. area"

QuickMergesort

Good candidate for X: ***Mergesort***

- #comparisons almost optimal
- but: merging uses extra space as buffer
- merging possible without extra space
  - but not really practical!



"Worst merge lanes in the D.C. area"

QuickMergesort

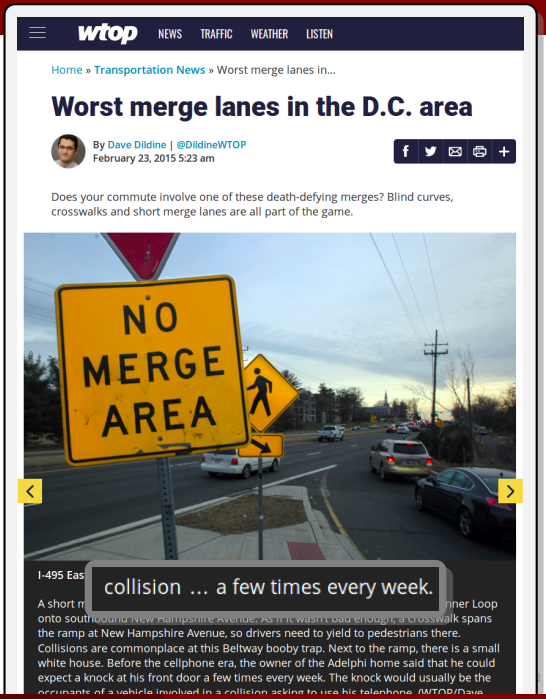
Good candidate for X: ***Mergesort***

- #comparisons almost optimal
- but: merging uses extra space as buffer
- merging possible without extra space
  - but not really practical!

For this talk: X = Mergesort

Goal: inherit good comparison count, but in-place!

"Worst merge lanes in the D.C. area"



QuickMergesort

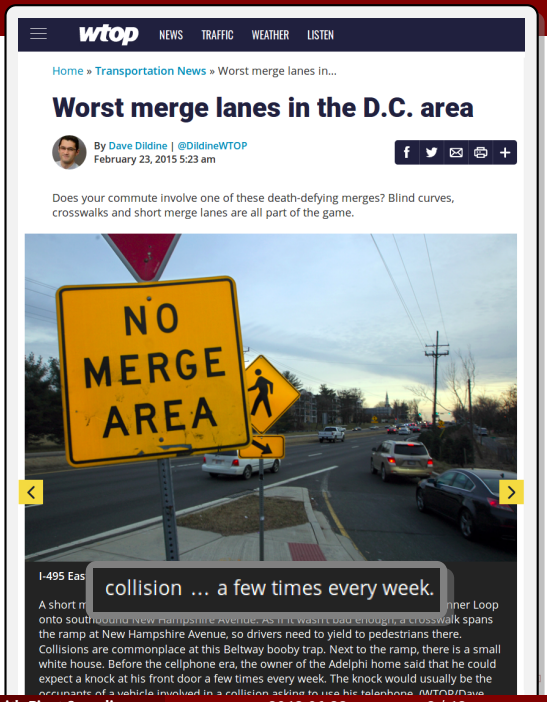
Good candidate for X: ***Mergesort***

- #comparisons almost optimal
- but: merging uses extra space as buffer
- merging possible without extra space
  - but not really practical!

For this talk: X = Mergesort

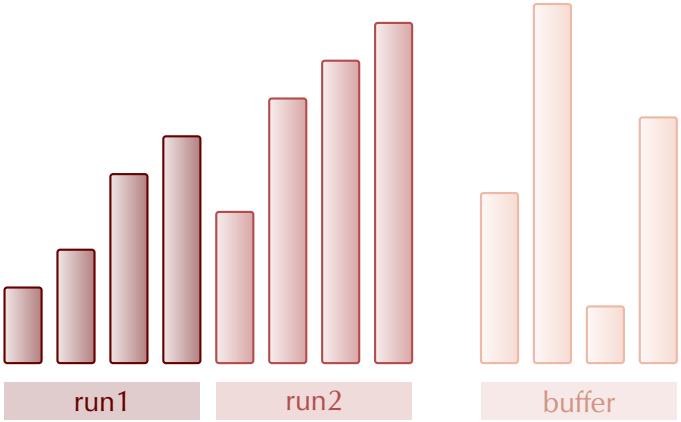
Goal: inherit good comparison count, but in-place! (and without collisions …)

"Worst merge lanes in the D.C. area"



Merging with swaps

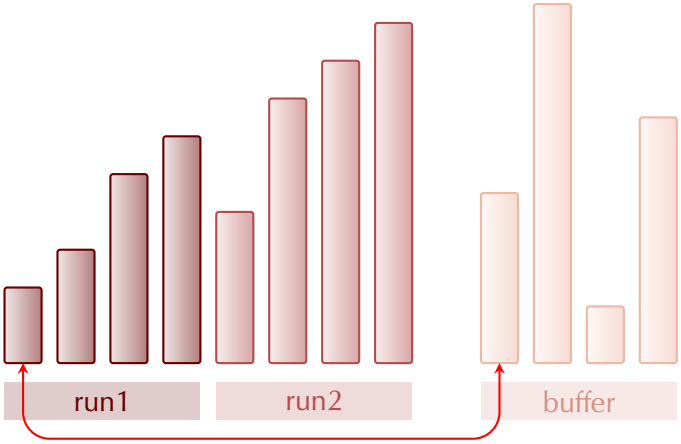
Important: *Buffer elements must stay intact.*



1. Swap run1 and buffer
2. Merge back (swap min)
3. runs merged, buffer restored (but permuted)

Merging with swaps

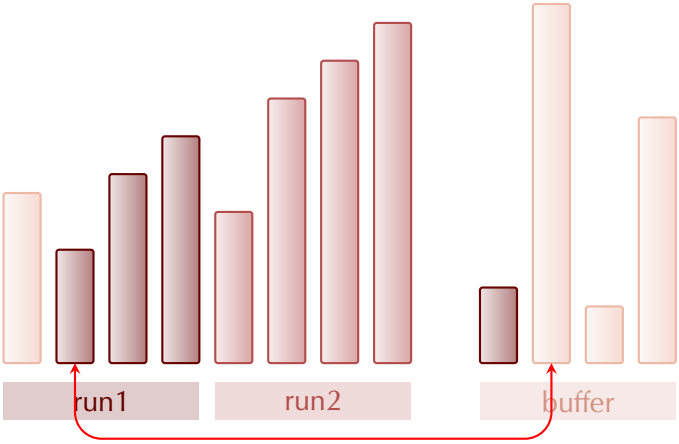
Important: *Buffer elements must stay intact.*



1. Swap run1 and buffer
2. Merge back (swap min)
3. runs merged, buffer restored (but permuted)

Merging with swaps

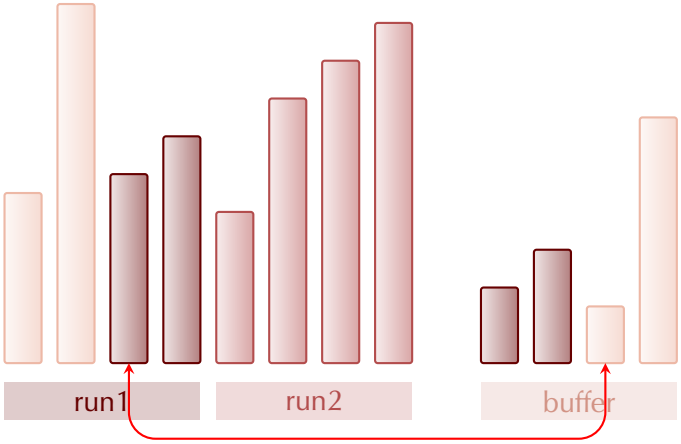
Important: *Buffer elements must stay intact.*



1. Swap run1 and buffer
2. Merge back (swap min)
3. runs merged, buffer restored (but permuted)

Merging with swaps

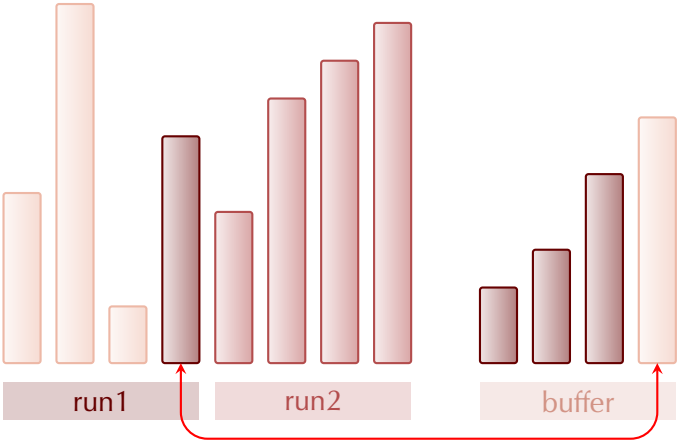
Important: *Buffer elements must stay intact.*



1. Swap run1 and buffer
2. Merge back (swap min)
3. runs merged, buffer restored (but permuted)

Merging with swaps

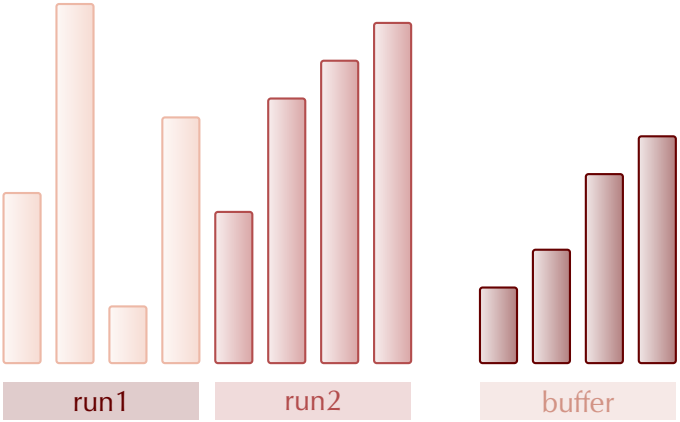
Important: *Buffer elements must stay intact.*



1. Swap run1 and buffer
2. Merge back (swap min)
3. runs merged, buffer restored (but permuted)

Merging with swaps

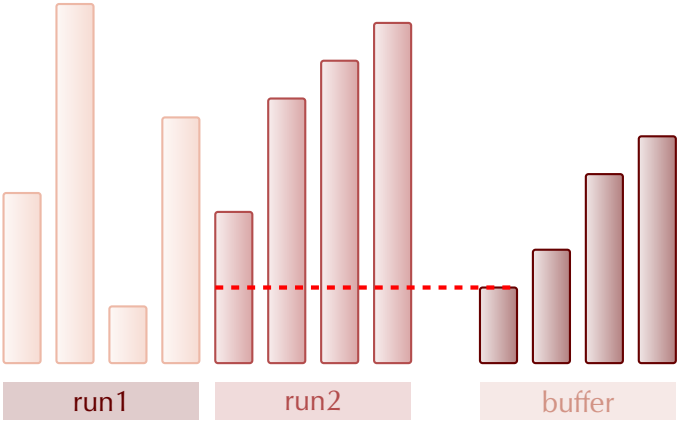
Important: *Buffer elements must stay intact.*



1. Swap run1 and buffer
2. Merge back (swap min)
3. runs merged, buffer restored (but permuted)

Merging with swaps

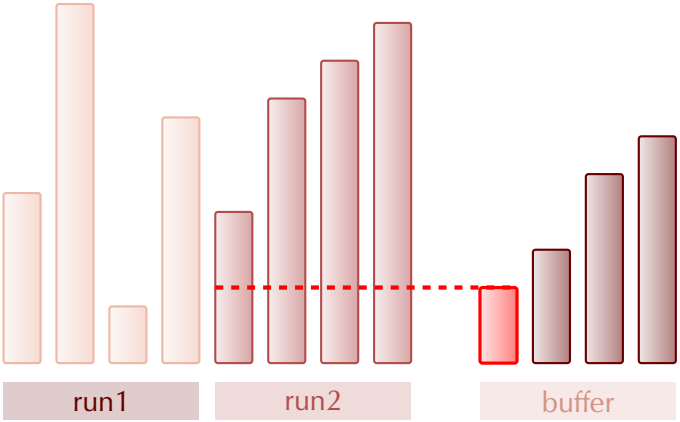
Important: *Buffer elements must stay intact.*



1. Swap run1 and buffer
2. Merge back (swap min)
3. runs merged, buffer restored (but permuted)

Merging with swaps

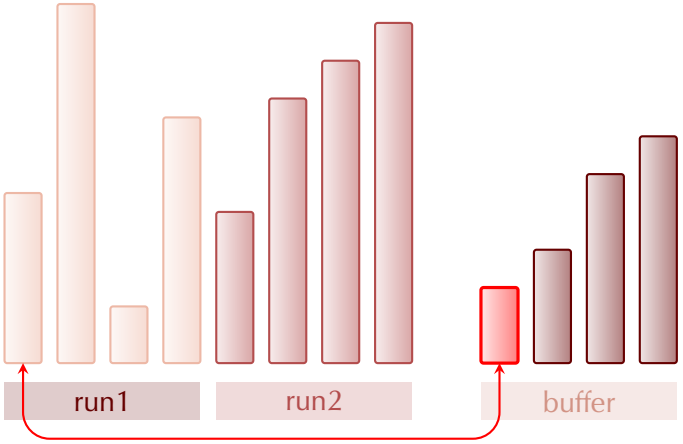
Important: *Buffer elements must stay intact.*



1. Swap run1 and buffer
2. Merge back (swap min)
3. runs merged, buffer restored (but permuted)

Merging with swaps

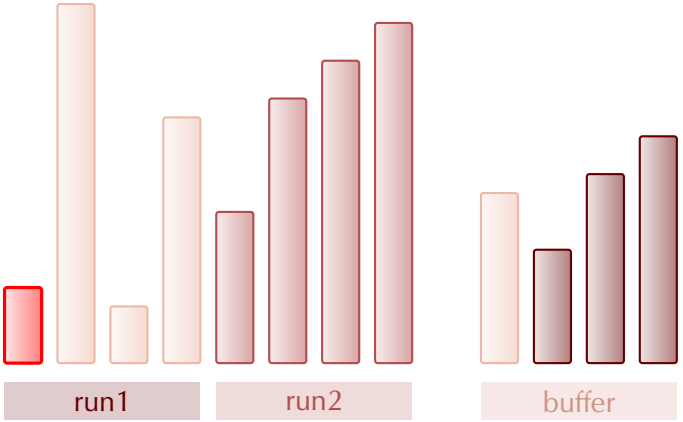
Important: *Buffer elements must stay intact.*



1. Swap run1 and buffer
2. Merge back (swap min)
3. runs merged, buffer restored (but permuted)

Merging with swaps

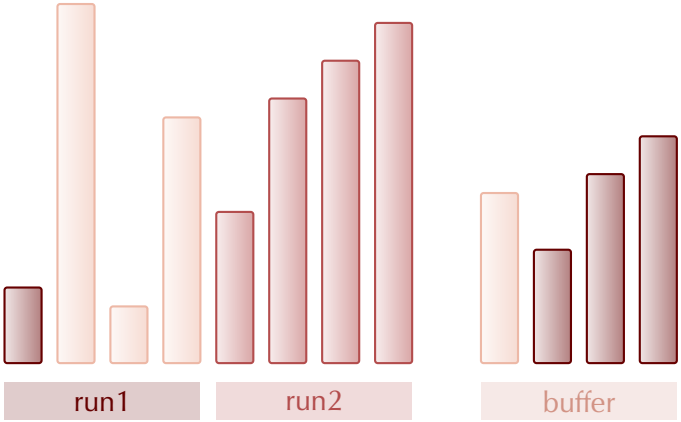
Important: *Buffer elements must stay intact.*



1. Swap run1 and buffer
2. Merge back (swap min)
3. runs merged, buffer restored (but permuted)

Merging with swaps

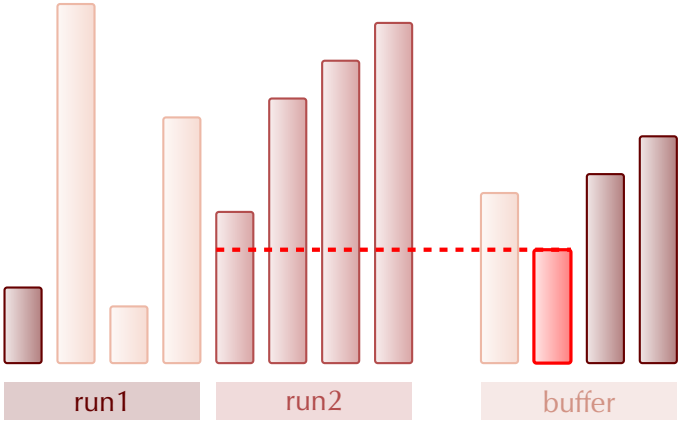
Important: *Buffer elements must stay intact.*



1. Swap run1 and buffer
2. Merge back (swap min)
3. runs merged, buffer restored (but permuted)

Merging with swaps

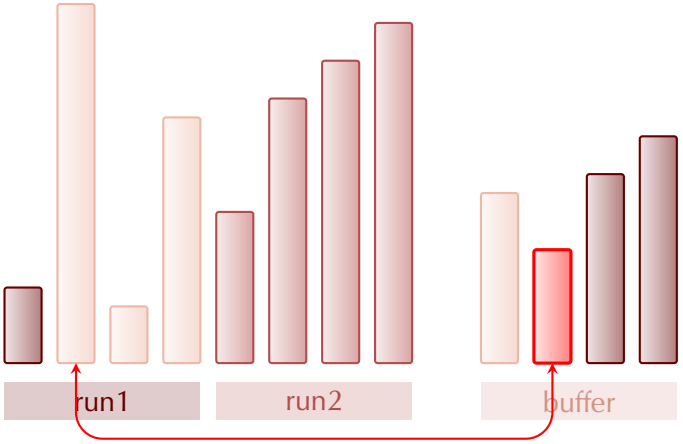
Important: *Buffer elements must stay intact.*



1. Swap run1 and buffer
2. Merge back (swap min)
3. runs merged, buffer restored (but permuted)

Merging with swaps

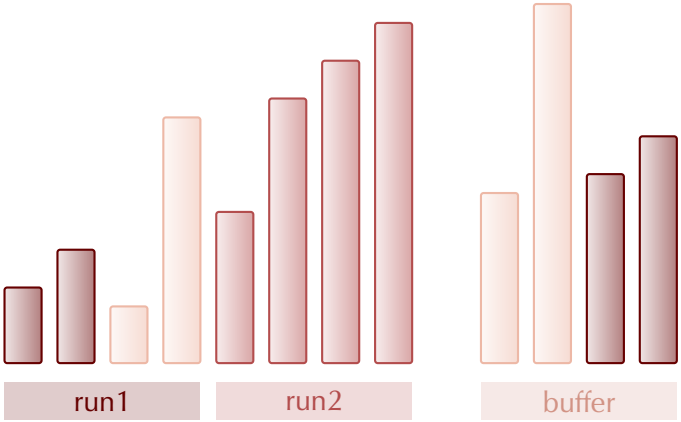
Important: *Buffer elements must stay intact.*



1. Swap run1 and buffer
2. Merge back (swap min)
3. runs merged, buffer restored (but permuted)

Merging with swaps

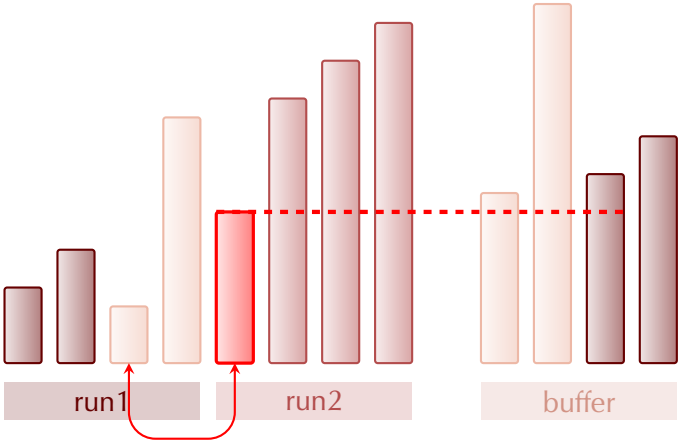
Important: *Buffer elements must stay intact.*



1. Swap run1 and buffer
2. Merge back (swap min)
3. runs merged, buffer restored (but permuted)

Merging with swaps

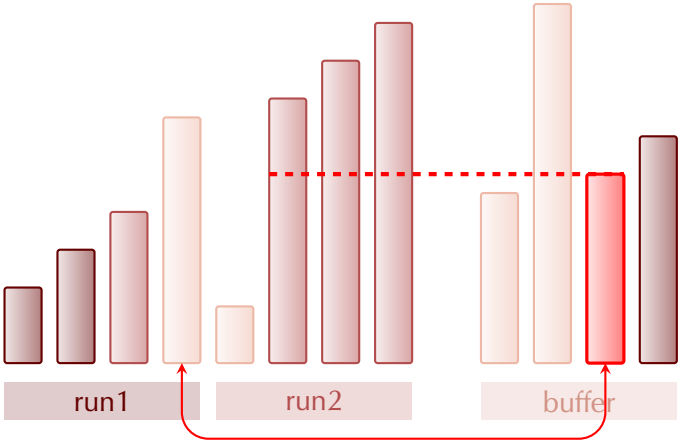
Important: *Buffer elements must stay intact.*



1. Swap run1 and buffer
2. Merge back (swap min)
3. runs merged, buffer restored (but permuted)

Merging with swaps

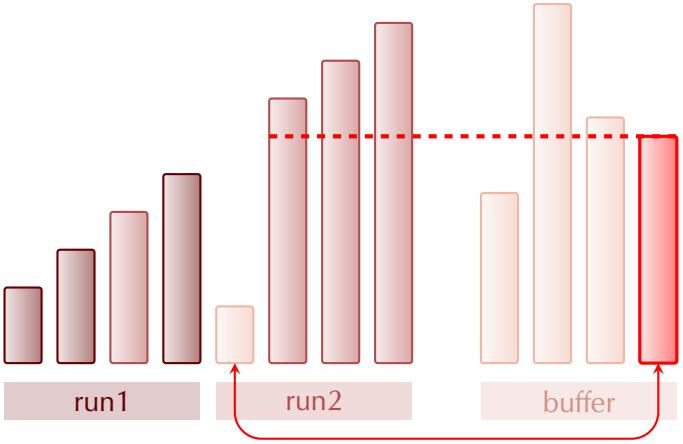
Important: *Buffer elements must stay intact.*



1. Swap run1 and buffer
2. Merge back (swap min)
3. runs merged, buffer restored (but permuted)

Merging with swaps

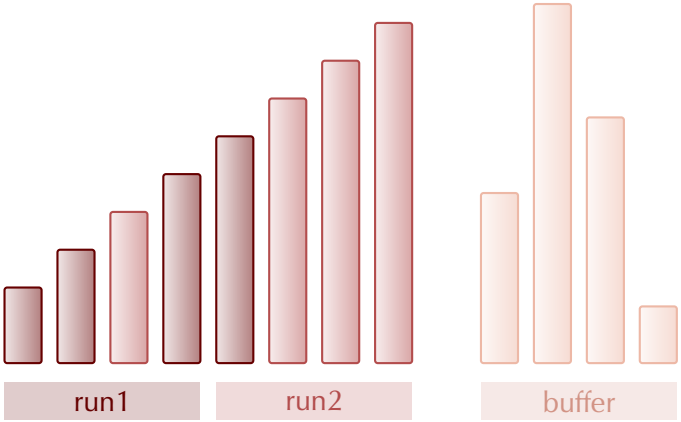
Important: *Buffer elements must stay intact.*



1. Swap run1 and buffer
2. Merge back (swap min)
3. runs merged, buffer restored (but permuted)

Merging with swaps

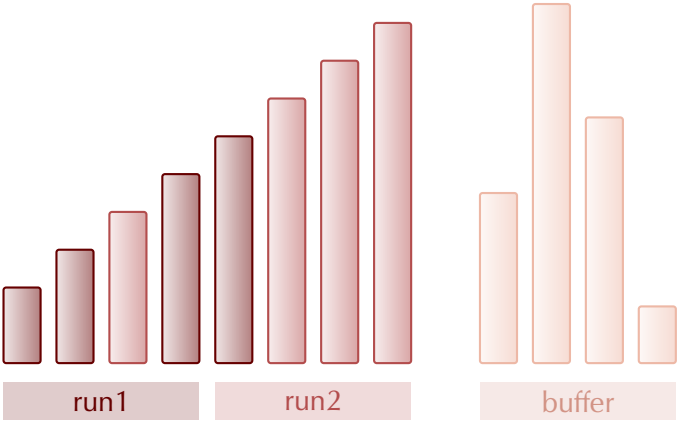
Important: *Buffer elements must stay intact.*



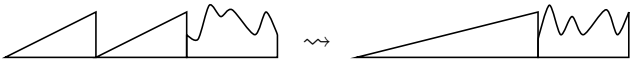
1. Swap run1 and buffer
2. Merge back (swap min)
3. runs merged, buffer restored (but permuted)

Merging with swaps

Important: *Buffer elements must stay intact.*



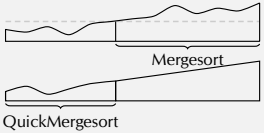
1. Swap run1 and buffer
2. Merge back (swap min)
3. runs merged, buffer restored (but permuted)



Which side with Mergesort?

QuickMergesort($A[l..r]$)

```
1  Choose pivot P.
2  i = Partition(A[l..r], P)
5  if ...
6      Mergesort(A[i+1..r]) with buffer A[l..i-1]
7      QuickMergesort(A[l..i-1]).
8  else
9      Mergesort(A[l..i-1]) with buffer A[i+1..r]
10     QuickMergesort(A[i..r]).
11 end if
```

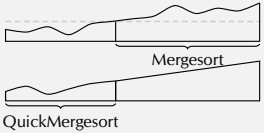


Which side with Mergesort?

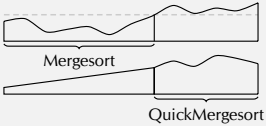
QuickMergesort($A[l..r]$)

1 Choose pivot P.
2 i = Partition($A[l..r]$, P)
3 $J_1 := i - l$; $J_2 := r - i$

5 **if** ...
6 MergeSort($A[i+1..r]$) with buffer $A[l..i-1]$
7 QuickMergesort($A[l..i-1]$).



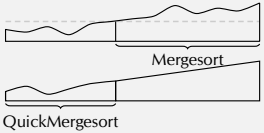
8 **else**
9 MergeSort($A[l..i-1]$) with buffer $A[i+1..r]$
10 QuickMergesort($A[i..r]$).
11 **end if**



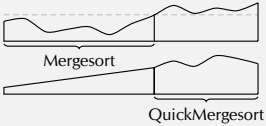
Which side with Mergesort?

QuickMergesort$(A[l..r])$

1 Choose pivot P.
2 $i = \text{Partition}(A[l..r], P)$
3 $J_1 := i - l; \quad J_2 := r - i$
4 *largerWithX* $:= \min\{J_1, J_2\} \geqslant \left\lfloor \frac{1}{2} \max\{J_1, J_2\} \right\rfloor$
5 **if** ...
6 Mergesort$(A[i+1..r])$ with buffer $A[l..i-1]$
7 QuickMergesort$(A[l..i-1])$.



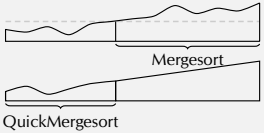
8 **else**
9 Mergesort$(A[l..i-1])$ with buffer $A[i+1..r]$
10 QuickMergesort$(A[i..r])$.
11 **end if**



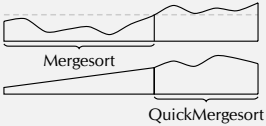
Which side with Mergesort?

QuickMergesort$(A[l..r])$

1 Choose pivot P.
2 i = Partition$(A[l..r], P)$
3 $J_1 := i - l$; $J_2 := r - i$
4 *largerWithX* $:= \min\{J_1, J_2\} \geqslant \left\lfloor \frac{1}{2} \max\{J_1, J_2\} \right\rfloor$
5 **if** $(\textit{largerWithX} \wedge J_2 \geqslant J_1)$
6 Mergesort$(A[i+1..r])$ with buffer $A[l..i-1]$
7 QuickMergesort$(A[l..i-1])$.



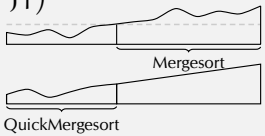
8 **else**
9 Mergesort$(A[l..i-1])$ with buffer $A[i+1..r]$
10 QuickMergesort$(A[i..r])$.
11 **end if**



Which side with Mergesort?

QuickMergesort$(A[l..r])$

1 Choose pivot P.
2 $i = \text{Partition}(A[l..r], P)$
3 $J_1 := i - l; \quad J_2 := r - i$
4 $\mathit{largerWithX} := \min\{J_1, J_2\} \geqslant \left\lfloor \frac{1}{2} \max\{J_1, J_2\} \right\rfloor$
5 **if** $(\mathit{largerWithX} \wedge J_2 \geqslant J_1) \ \vee \ (\neg \mathit{largerWithX} \wedge J_2 < J_1)$
6 Mergesort$(A[i+1..r])$ with buffer $A[l..i-1]$
7 QuickMergesort$(A[l..i-1])$.
8 **else**
9 Mergesort$(A[l..i-1])$ with buffer $A[i+1..r]$
10 QuickMergesort$(A[i..r])$.
11 **end if**



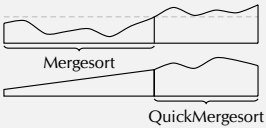
Which side with Mergesort?

QuickMergesort$(A[l..r])$

1 Choose pivot P.
2 i = Partition$(A[l..r], P)$
3 $J_1 := i - l$; $J_2 := r - i$
4 *largerWithX* $:= \min\{J_1, J_2\} \geqslant \left\lfloor \frac{1}{2} \max\{J_1, J_2\} \right\rfloor$
5 **if** $\big($*largerWithX* $\wedge\ J_2 \geqslant J_1\big)\ \vee\ \big(\neg$*largerWithX* $\wedge\ J_2$ <
6 Mergesort$(A[i+1..r])$ with buffer $A[l..i-1]$
7 QuickMergesort$(A[l..i-1])$.
8 **else**
9 Mergesort$(A[l..i-1])$ with buffer $A[i+1..r]$
10 QuickMergesort$(A[i..r])$.
11 **end if**

Sort **recursively** iff **relative size** fulfills

$$\frac{J_1}{n-1} \in \left[\frac{1}{3}, \frac{1}{2}\right] \cup \left(\frac{2}{3}, 1\right]$$



Which side with Mergesort?

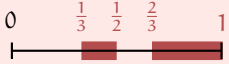
QuickMergesort$(A[l..r])$

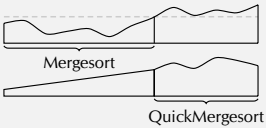
1 Choose pivot P.
2 $i = \text{Partition}(A[l..r], P)$
3 $J_1 := i - l; \quad J_2 := r - i$
4 $\mathit{largerWithX} := \min\{J_1, J_2\} \geqslant \left\lfloor \frac{1}{2} \max\{J_1, J_2\} \right\rfloor$
5 **if** $(\mathit{largerWithX} \wedge J_2 \geqslant J_1) \ \vee \ (\neg \mathit{largerWithX} \wedge J_2$
6 Mergesort$(A[i+1..r])$ with buffer $A[l..i-1]$
7 QuickMergesort$(A[l..i-1])$.
8 **else**
9 Mergesort$(A[l..i-1])$ with buffer $A[i+1..r]$
10 QuickMergesort$(A[i..r])$.
11 **end if**

Sort **recursively** iff **relative size** fulfills

$$\frac{J_1}{n-1} \in \left[\frac{1}{3}, \frac{1}{2}\right] \cup \left(\frac{2}{3}, 1\right]$$

$$\frac{J_1}{n-1} \in$$





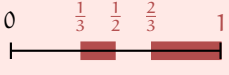
Which side with Mergesort?

QuickMergesort$(A[l..r])$

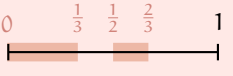
1 Choose pivot P.
2 $i = \text{Partition}(A[l..r], P)$
3 $J_1 := i - l$; $J_2 := r - i$
4 *largerWithX* $:= \min\{J_1, J_2\} \geqslant \left\lfloor \frac{1}{2} \max\{J_1, J_2\} \right\rfloor$
5 **if** $\big($*largerWithX* $\wedge\ J_2 \geqslant J_1\big) \ \vee\ \big(\neg$*largerWithX* $\wedge\ J_2$ <
6 MergeSort$(A[i+1..r])$ with buffer $A[l..i-1]$
7 QuickMergesort$(A[l..i-1])$.
8 **else**
9 MergeSort$(A[l..i-1])$ with buffer $A[i+1..r]$
10 QuickMergesort$(A[i..r])$.
11 **end if**

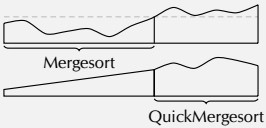
Sort **recursively** iff **relative size** fulfills

$$\frac{J_1}{n-1} \in \left[\frac{1}{3}, \frac{1}{2}\right] \cup \left(\frac{2}{3}, 1\right]$$



Similarly: **mergesort** iff





Outline

1 What is QuickXsort?

2 Recurrence

3 Toll function

4 Result

Costs of QuickXsort

Assumptions: #comparisons

- average costs for X: $x(n) = an \lg n + bn \pm O(n^{1-\varepsilon})$ — αn buffer space
- X = Mergesort: $x(n) = n \lg n + [-1.26, -1.24]n \pm O(1)$ — $\frac{1}{2}n$ buffer space

Costs of QuickXsort

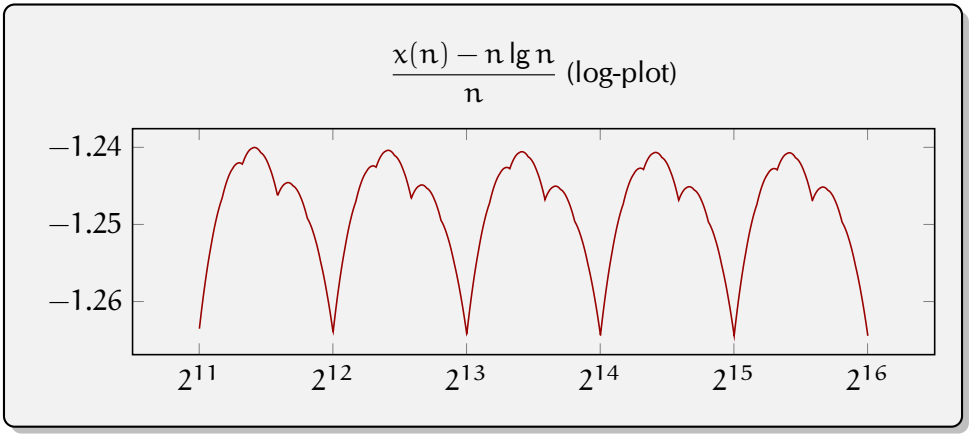
Assumptions: #comparisons

- average costs for X: $x(n) = an \lg n + bn \pm O(n^{1-\varepsilon})$ — αn buffer space
- X = Mergesort: $x(n) = n \lg n + [-1.26, -1.24]n \pm O(1)$ — $\frac{1}{2}n$ buffer space

Costs of QuickXsort

Assumptions: #comparisons

- average costs for X: $x(n) = an \lg n + bn \pm O(n^{1-\varepsilon})$ — αn buffer space
- X = Mergesort: $x(n) = n \lg n + [-1.26, -1.24]n \pm O(1)$ — $\frac{1}{2}n$ buffer space



Costs of QuickXsort

Assumptions: #comparisons

- average costs for X: $x(n) = an \lg n + bn \pm O(n^{1-\varepsilon})$ — αn buffer space
- X = Mergesort: $x(n) = n \lg n + [-1.26, -1.24]n \pm O(1)$ — $\frac{1}{2}n$ buffer space

Costs of QuickXsort

Assumptions: #comparisons

- average costs for X: $x(n) = an\lg n + bn \pm O(n^{1-\varepsilon})$ — αn buffer space
- X = Mergesort: $x(n) = n\lg n + [-1.26, -1.24]n \pm O(1)$ — $\frac{1}{2}n$ buffer space

$$c(n) = (n-k) + b(k) + \mathbb{E}\left[\left[\tfrac{J_1}{n-1} \in \left[\tfrac{1}{3},\tfrac{1}{2}\right] \cup \left(\tfrac{2}{3},1\right]\right] \cdot \left(x(J_2) + c(J_1)\right)\right]$$
$$+ \mathbb{E}\left[\left[\tfrac{J_2}{n-1} \in \left[\tfrac{1}{3},\tfrac{1}{2}\right) \cup \left(\tfrac{2}{3},1\right]\right] \cdot \left(x(J_1) + c(J_2)\right)\right] \qquad (n > k)$$

Costs of QuickXsort

Assumptions: (#comparisons)

- average costs for X: $x(n) = an\lg n + bn \pm O(n^{1-\varepsilon})$ — αn buffer space
- X = Mergesort: $x(n) = n\lg n + [-1.26, -1.24]n \pm O(1)$ — $\frac{1}{2}n$ buffer space

(partition)

$$c(n) = (n-k) + b(k) + \mathbb{E}\left[\left[\tfrac{J_1}{n-1} \in \left[\tfrac{1}{3},\tfrac{1}{2}\right] \cup \left(\tfrac{2}{3},1\right]\right] \cdot \left(x(J_2) + c(J_1)\right)\right] + \mathbb{E}\left[\left[\tfrac{J_2}{n-1} \in \left[\tfrac{1}{3},\tfrac{1}{2}\right) \cup \left(\tfrac{2}{3},1\right]\right] \cdot \left(x(J_1) + c(J_2)\right)\right] \qquad (n > k)$$

Costs of QuickXsort

Assumptions: (#comparisons)

- average costs for X: $x(n) = an\lg n + bn \pm O(n^{1-\varepsilon})$ — αn buffer space
- X = Mergesort: $x(n) = n\lg n + [-1.26, -1.24]n \pm O(1)$ — $\frac{1}{2}n$ buffer space

(pivot sampling)

$$c(n) = (n-k) + b(k) + \mathbb{E}\left[\left[\tfrac{J_1}{n-1} \in \left[\tfrac{1}{3},\tfrac{1}{2}\right] \cup \left(\tfrac{2}{3},1\right]\right] \cdot \left(x(J_2) + c(J_1)\right)\right]$$
$$+ \mathbb{E}\left[\left[\tfrac{J_2}{n-1} \in \left[\tfrac{1}{3},\tfrac{1}{2}\right) \cup \left(\tfrac{2}{3},1\right]\right] \cdot \left(x(J_1) + c(J_2)\right)\right] \qquad (n > k)$$

Costs of QuickXsort

Assumptions: (#comparisons)

- average costs for X: $x(n) = an\lg n + bn \pm O(n^{1-\varepsilon})$ — αn buffer space
- X = Mergesort: $x(n) = n\lg n + [-1.26, -1.24]n \pm O(1)$ — $\frac{1}{2}n$ buffer space

(Mergesort right; recurse left)

$$c(n) = (n-k) + b(k) + \mathbb{E}\left[\left[\tfrac{J_1}{n-1} \in \left[\tfrac{1}{3},\tfrac{1}{2}\right] \cup \left(\tfrac{2}{3},1\right]\right] \cdot \big(x(J_2) + c(J_1)\big)\right]$$
$$+ \mathbb{E}\left[\left[\tfrac{J_2}{n-1} \in \left[\tfrac{1}{3},\tfrac{1}{2}\right) \cup \left(\tfrac{2}{3},1\right]\right] \cdot \big(x(J_1) + c(J_2)\big)\right] \qquad (n > k)$$

Costs of QuickXsort

Assumptions: (#comparisons)

- average costs for X: $x(n) = an\lg n + bn \pm O(n^{1-\varepsilon})$ — αn buffer space
- X = Mergesort: $x(n) = n\lg n + [-1.26, -1.24]n \pm O(1)$ — $\frac{1}{2}n$ buffer space

$$c(n) = (n-k) + b(k) + \mathbb{E}\left[\left[\tfrac{J_1}{n-1} \in \left[\tfrac{1}{3},\tfrac{1}{2}\right] \cup \left(\tfrac{2}{3},1\right]\right] \cdot \big(x(J_2) + c(J_1)\big)\right] + \mathbb{E}\left[\left[\tfrac{J_2}{n-1} \in \left[\tfrac{1}{3},\tfrac{1}{2}\right) \cup \left(\tfrac{2}{3},1\right]\right] \cdot \big(x(J_1) + c(J_2)\big)\right] \qquad (n > k)$$

Mergesort left — recurse right

Costs of QuickXsort

Assumptions: #comparisons

- average costs for X: $x(n) = an\lg n + bn \pm O(n^{1-\varepsilon})$ — αn buffer space
- X = Mergesort: $x(n) = n\lg n + [-1.26, -1.24]n \pm O(1)$ — $\frac{1}{2}n$ buffer space

$$c(n) = (n-k) + b(k) + \mathbb{E}\left[\left[\tfrac{J_1}{n-1} \in \quad\right] \cdot \big(x(J_2) + c(J_1)\big)\right] + \mathbb{E}\left[\left[\tfrac{J_2}{n-1} \in \quad\right] \cdot \big(x(J_1) + c(J_2)\big)\right] \qquad (n > k)$$

Costs of QuickXsort

Assumptions: #comparisons

- average costs for X: $x(n) = an\lg n + bn \pm O(n^{1-\varepsilon})$ — αn buffer space
- X = Mergesort: $x(n) = n\lg n + [-1.26, -1.24]n \pm O(1)$ — $\frac{1}{2}n$ buffer space

$$c(n) = (n-k) + b(k) + \mathbb{E}\left[\left[\tfrac{J_1}{n-1} \in \quad\right] \cdot \big(x(J_2) + c(J_1)\big)\right] + \mathbb{E}\left[\left[\tfrac{J_2}{n-1} \in \quad\right] \cdot \big(x(J_1) + c(J_2)\big)\right] \qquad (n > k)$$



+ base cases

Costs of QuickXsort

Assumptions: #comparisons

- average costs for X: $x(n) = an \lg n + bn \pm O(n^{1-\varepsilon})$ — αn buffer space
- X = Mergesort: $x(n) = n \lg n + [-1.26, -1.24]n \pm O(1)$ — $\frac{1}{2}n$ buffer space

$$
\begin{aligned}
c(n) &= (n-k) + b(k) + \mathbb{E}\left[\left[\tfrac{J_1}{n-1} \in \right] \cdot \big(x(J_2) + c(J_1)\big)\right] \\
&\qquad + \mathbb{E}\left[\left[\tfrac{J_2}{n-1} \in \right] \cdot \big(x(J_1) + c(J_2)\big)\right] \qquad (n > k) \\
&= t(n) + \sum_{j=0}^{n-1} w_{n,j} \cdot c(j)
\end{aligned}
$$

Costs of QuickXsort

Assumptions: #comparisons

- average costs for X: $x(n) = an\lg n + bn \pm O(n^{1-\varepsilon})$ — αn buffer space
- X = Mergesort: $x(n) = n\lg n + [-1.26, -1.24]n \pm O(1)$ — $\frac{1}{2}n$ buffer space

$$c(n) = (n-k) + b(k) + \mathbb{E}\left[\left[\frac{J_1}{n-1} \in$$

$$+ \mathbb{E}\left[\left[\frac{J_2}{n-1} \in$$

$$= t(n) + \sum_{j=0}^{n-1} w_{n,j} \cdot c(j)$$

shape for Roura's ***continuous master theorem***:

- $c(n)$ not dominated by "leaves" of recursion
- $t(n)$ "standard scale": $t(n) \sim Kn^{\alpha}\log^{\beta} n$
- $w_{n,j}$ "smooth": $w_{n,zn} \approx w(z)$

$\rightsquigarrow$ $c(n) \sim$ solution of $c(n) = t(n) + \int w(z)c(zn)dz$
(which is known explicitly)

Costs of QuickXsort

Assumptions: #comparisons

- average costs for X: $x(n) = an \lg n + bn \pm O(n^{1-\varepsilon})$ — αn buffer space
- X = Mergesort: $x(n) = n \lg n + [-1.26, -1.24]n \pm O(1)$ — $\frac{1}{2}n$ buffer space

$$c(n) = (n-k) + b(k) + \mathbb{E}\left[\left[\frac{J_1}{n-1} \in$$

$$+ \mathbb{E}\left[\left[\frac{J_2}{n-1} \in$$

$$= t(n) + \sum_{j=0}^{n-1} w_{n,j} \cdot c(j)$$

shape for Roura's ***continuous master theorem***:

- $c(n)$ not dominated by "leaves" of recursion
- $t(n)$ "standard scale": $t(n) \sim Kn^{\alpha} \log^{\beta} n$
- $w_{n,j}$ "smooth": $w_{n,zn} \approx w(z)$

$\rightsquigarrow$ $c(n) \sim$ solution of $c(n) = t(n) + \int w(z)c(zn)dz$
(which is known explicitly)

Costs of QuickXsort

Assumptions: #comparisons

- average costs for X: $x(n) = an\lg n + bn \pm O(n^{1-\varepsilon})$ — αn buffer space
- X = Mergesort: $x(n) = n\lg n + [-1.26, -1.24]n \pm O(1)$ — $\frac{1}{2}n$ buffer space

$$c(n) = (n-k) + b(k) + \mathbb{E}\left[\left[\frac{J_1}{n-1} \in$$

$$+ \mathbb{E}\left[\left[\frac{J_2}{n-1} \in$$

$$= t(n) + \sum_{j=0}^{n-1} w_{n,j}\cdot c(j)$$

shape for Roura's ***continuous master theorem***:

- $c(n)$ not dominated by "leaves" of recursion
- $t(n)$ "standard scale": $t(n) \sim Kn^{\alpha}\log^{\beta} n$
- $w_{n,j}$ "smooth": $w_{n,zn} \approx w(z)$

$\rightsquigarrow$ $c(n) \sim$ solution of $c(n) = t(n) + \int w(z)c(zn)dz$
(which is known explicitly)

Costs of QuickXsort

Assumptions: #comparisons

- average costs for X: $x(n) = an\lg n + bn \pm O(n^{1-\varepsilon})$ — αn buffer space
- X = Mergesort: $x(n) = n\lg n + [-1.26, -1.24]n \pm O(1)$ — $\frac{1}{2}n$ buffer space

$$c(n) = (n-k) + b(k) + \mathbb{E}\left[\left[\frac{J_1}{n-1} \in$$

$$+ \mathbb{E}\left[\left[\frac{J_2}{n-1} \in$$

$$= t(n) + \sum_{j=0}^{n-1} w_{n,j}\cdot c(j)$$

shape for Roura's ***continuous master theorem***:

- $c(n)$ not dominated by "leaves" of recursion
- $t(n)$ "standard scale": $t(n) \sim Kn^{\alpha}\log^{\beta} n$
- $w_{n,j}$ "smooth": $w_{n,zn} \approx w(z)$

$\rightsquigarrow$ $c(n) \sim$ solution of $c(n) = t(n) + \int w(z)c(zn)dz$
(which is known explicitly)

Costs of QuickXsort

Assumptions: (#comparisons)

- average costs for X: $x(n) = an\lg n + bn \pm O(n^{1-\varepsilon})$ — αn buffer space
- X = Mergesort: $x(n) = n\lg n + [-1.26, -1.24]n \pm O(1)$ — $\frac{1}{2}n$ buffer space

$$c(n) = (n-k) + b(k) + \mathbb{E}\left[\left[\frac{J_1}{n-1} \in$$

$$+ \mathbb{E}\left[\left[\frac{J_2}{n-1} \in$$

$$= t(n) + \sum_{j=0}^{n-1} w_{n,j}\cdot c(j)$$

shape for Roura's ***continuous master theorem***:

- $c(n)$ not dominated by "leaves" of recursion
- $t(n)$ "standard scale": $t(n) \sim Kn^{\alpha}\log^{\beta} n$
- $w_{n,j}$ "smooth": $w_{n,zn} \approx w(z)$

$\rightsquigarrow$ $c(n) \sim$ solution of $c(n) = t(n) + \int w(z)c(zn)dz$
(which is known explicitly)

Costs of QuickXsort

Assumptions: #comparisons

- average costs for X: $x(n) = an\lg n + bn \pm O(n^{1-\varepsilon})$ — αn buffer space
- X = Mergesort: $x(n) = n\lg n + [-1.26, -1.24]n \pm O(1)$ — $\frac{1}{2}n$ buffer space

$$c(n) = (n-k) + b(k) + \mathbb{E}\left[\left[\frac{J_1}{n-1} \in$$

$$+ \mathbb{E}\left[\left[\frac{J_2}{n-1} \in$$

$$= t(n) + \sum_{j=0}^{n-1} w_{n,j}\cdot c(j)$$

shape for Roura's ***continuous master theorem***:

- $c(n)$ not dominated by "leaves" of recursion ✓
- $t(n)$ "standard scale": $t(n) \sim Kn^{\alpha}\log^{\beta} n$
- $w_{n,j}$ "smooth": $w_{n,zn} \approx w(z)$

$\rightsquigarrow$ $c(n) \sim$ solution of $c(n) = t(n) + \int w(z)c(zn)dz$
(which is known explicitly)

Costs of QuickXsort

Assumptions: (#comparisons)

- average costs for X: $x(n) = an\lg n + bn \pm O(n^{1-\varepsilon})$ — αn buffer space
- X = Mergesort: $x(n) = n\lg n + [-1.26, -1.24]n \pm O(1)$ — $\frac{1}{2}n$ buffer space

$$c(n) = (n-k) + b(k) + \mathbb{E}\left[\left[\frac{J_1}{n-1} \in \right.\right. + \mathbb{E}\left[\left[\frac{J_2}{n-1} \in \right.\right.)$$

$$= t(n) + \sum_{j=0}^{n-1} w_{n,j}\cdot c(j)$$

shape for Roura's ***continuous master theorem***:

- $c(n)$ not dominated by "leaves" of recursion ✓
- $t(n)$ "standard scale": $t(n) \sim Kn^{\alpha}\log^{\beta} n$ (✓)
- $w_{n,j}$ "smooth": $w_{n,zn} \approx w(z)$

$\rightsquigarrow$ $c(n) \sim$ solution of $c(n) = t(n) + \int w(z)c(zn)dz$
(which is known explicitly)

$$t(n) = (n-k) + b(k) + \mathbb{E}\left[\left[\frac{J_1}{n-1} \in \right]x(J_1)\right] + \mathbb{E}\left[\left[\frac{J_2}{n-1} \in \right]x(J_2)\right]$$

$$w_{n,j} = \begin{cases} 1\cdot\mathbb{P}[J_1 = j] & \text{if } \frac{j}{n-1} = \frac{1}{2} \\ 2\cdot\mathbb{P}[J_1 = j] & \text{if } \frac{j}{n-1} \in \\ 0 & \text{otherwise} \end{cases}$$

Costs of QuickXsort

Assumptions: (#comparisons)

- average costs for X: $x(n) = an\lg n + bn \pm O(n^{1-\varepsilon})$ — αn buffer space
- X = Mergesort: $x(n) = n\lg n + [-1.26, -1.24]n \pm O(1)$ — $\frac{1}{2}n$ buffer space

$$c(n) = (n-k) + b(k) + \mathbb{E}\left[\left[\frac{J_1}{n-1} \in \right.\right. + \mathbb{E}\left[\left[\frac{J_2}{n-1} \in \right.\right.$$

$$= t(n) + \sum_{j=0}^{n-1} w_{n,j}\cdot c(j)$$

shape for Roura's ***continuous master theorem***:

- $c(n)$ not dominated by "leaves" of recursion ✓
- $t(n)$ "standard scale": $t(n) \sim Kn^{\alpha}\log^{\beta} n$ (✓)
- $w_{n,j}$ "smooth": $w_{n,zn} \approx w(z)$ (✓)

$\rightsquigarrow$ $c(n) \sim$ solution of $c(n) = t(n) + \int w(z)c(zn)dz$
(which is known explicitly)

$$t(n) = (n-k) + b(k) + \mathbb{E}\left[\left[\frac{J_1}{n-1} \in \right]x(J_1)\right] + \mathbb{E}\left[\left[\frac{J_2}{n-1} \in \right]x(J_2)\right]$$

$$w_{n,j} = \begin{cases} 1\cdot\mathbb{P}[J_1 = j] & \text{if } \frac{j}{n-1} = \frac{1}{2} \\ 2\cdot\mathbb{P}[J_1 = j] & \text{if } \frac{j}{n-1} \in \\ 0 & \text{otherwise} \end{cases}$$

Costs of QuickXsort

Assumptions: (#comparisons)

- average costs for X: $x(n) = an\lg n + bn \pm O(n^{1-\varepsilon})$ — αn buffer space
- X = Mergesort: $x(n) = n\lg n + [-1.26, -1.24]n \pm O(1)$ — $\frac{1}{2}n$ buffer space

$$c(n) = (n-k) + b(k) + \mathbb{E}\left[x(J_1) + x(J_2)\right]$$

$$= t(n) + \sum_{j=0}^{n-1} w_{n,j}\cdot c(j)$$

$$t(n) = (n-k) + b(k) + \mathbb{E}\left[x(J_1) + x(J_2)\right]$$

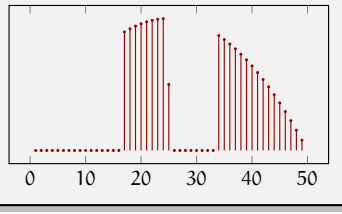
$$w_{n,j} = \begin{cases} 1\cdot\mathbb{P}[J_1 = j] & \text{if } \frac{j}{n-1} = \frac{1}{2} \\ 2\cdot\mathbb{P}[J_1 = j] & \text{if } \frac{j}{n-1} \in \\ 0 & \text{otherwise} \end{cases}$$

shape for Roura's ***continuous master theorem***:

- $c(n)$ not dominated by "leaves" of recursion ✓
- $t(n)$ "standard scale": $t(n) \sim Kn^{\alpha}\log^{\beta} n$ (✓)
- $w_{n,j}$ "smooth": $w_{n,zn} \approx w(z)$ (✓)

$\rightsquigarrow$ $c(n) \sim$ solution of $c(n) = t(n) + \int w(z)c(zn)dz$
(which is known explicitly)

$w_{51,j}$ for median-of-3 QuickMergesort



Shape Function

$w_{n,j}$ *"smooth"*

- precise criterion:

$$\exists w : [0,1] \to \mathbb{R}_{\geq 0} \;:\; \sum_{j=0}^{n-1} \left| w_{n,j} - \int_{j/n}^{(j+1)/n} w(z)\,\mathrm{d}z \right| \;=\; O(n^{-1}) \qquad (n \to \infty).$$

- subproblem size distribution: $J_1 \overset{\mathcal{D}}{=} J_2 \overset{\mathcal{D}}{=} \text{Binomial}(n, \text{Beta}(t+1, t+1)) \pm O(1)$

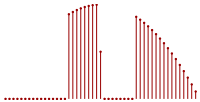
- binomial distribution highly concentrated $\rightsquigarrow$ $\frac{J_1}{n} \overset{\mathcal{D}}{\approx} P \overset{\mathcal{D}}{=} \text{Beta}(t+1, t+1)$
  formally: $n\mathbb{P}\big[J_1 = \lfloor zn \rfloor\big] = f_{\text{Beta}}(z) \pm O(n^{-1})$

$\rightsquigarrow$ $w(z) = 2\big[z \in \longmapsto\!\!\!

Shape Function

$w_{n,j}$ *"smooth"*

$$w_{n,j} = \begin{cases} 1 \cdot \mathbb{P}[J_1 = j] & \text{if } \frac{j}{n-1} = \frac{1}{2} \\ 2 \cdot \mathbb{P}[J_1 = j] & \text{if } \frac{j}{n-1} \in \vdash\!\!\blacksquare\!\!\dashv \\ 0 & \text{otherwise} \end{cases}$$



- precise criterion:

$$\exists w : [0,1] \to \mathbb{R}_{\geq 0} \;:\; \sum_{j=0}^{n-1} \left| w_{n,j} - \int_{j/n}^{(j+1)/n} w(z)\,dz \right| \;=\; O(n^{-1}) \qquad (n \to \infty).$$

the "shape function"

- subproblem size distribution: $J_1 \overset{\mathcal{D}}{=} J_2 \overset{\mathcal{D}}{=} \text{Binomial}(n, \text{Beta}(t+1, t+1)) \pm O(1)$

beta-binomial distribution

- binomial distribution highly concentrated $\rightsquigarrow$ $\frac{J_1}{n} \overset{\mathcal{D}}{\approx} P \overset{\mathcal{D}}{=} \text{Beta}(t+1, t+1)$

Chernoff etc.

median of $2t+1$ i.i.d. Uniform$[0,1]$ rv.

formally: $n\mathbb{P}\big[J_1 = \lfloor zn \rfloor\big] = f_{\text{Beta}}(z) \pm O(n^{-1})$

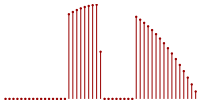
$= z^t(1-z)^t / B(t+1, t+1)$

$\rightsquigarrow$ $w(z) = 2\big[z \in \vdash\!\!\blacksquare\!\!\dashv\big] \cdot f_{\text{Beta}}(z)$ valid shape function

Shape Function

$w_{n,j}$ *"smooth"*

$$w_{n,j} = \begin{cases} 1 \cdot \mathbb{P}[J_1 = j] & \text{if } \frac{j}{n-1} = \frac{1}{2} \\ 2 \cdot \mathbb{P}[J_1 = j] & \text{if } \frac{j}{n-1} \in \vdash\!\!\blacksquare\!\!\dashv \\ 0 & \text{otherwise} \end{cases}$$



- precise criterion:

$$\exists w : [0,1] \to \mathbb{R}_{\geqslant 0} \;:\; \sum_{j=0}^{n-1} \left| w_{n,j} - \int_{j/n}^{(j+1)/n} w(z)\,dz \right| \;=\; O(n^{-1}) \qquad (n \to \infty).$$

the "shape function"

- subproblem size distribution: $J_1 \overset{\mathcal{D}}{=} J_2 \overset{\mathcal{D}}{=} \text{Binomial}(n, \text{Beta}(t+1, t+1)) \pm O(1)$

beta-binomial distribution

- binomial distribution highly concentrated $\rightsquigarrow$ $\frac{J_1}{n} \overset{\mathcal{D}}{\approx} P \overset{\mathcal{D}}{=} \text{Beta}(t+1, t+1)$

Chernoff etc.

median of $2t+1$ i.i.d. Uniform$[0,1]$ rv.

formally: $n\mathbb{P}[J_1 = \lfloor zn \rfloor] = f_{\text{Beta}}(z) \pm O(n^{-1})$

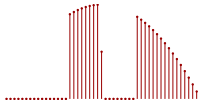
$= z^t(1-z)^t / B(t+1, t+1)$

$\rightsquigarrow$ $w(z) = 2\left[z \in \vdash\!\!\blacksquare\!\!\dashv\right] \cdot f_{\text{Beta}}(z)$ valid shape function

Shape Function

$w_{n,j}$ *"smooth"*

$$w_{n,j} = \begin{cases} 1 \cdot \mathbb{P}[J_1 = j] & \text{if } \frac{j}{n-1} = \frac{1}{2} \\ 2 \cdot \mathbb{P}[J_1 = j] & \text{if } \frac{j}{n-1} \in \vdash\!\!\blacksquare\!\!\dashv \\ 0 & \text{otherwise} \end{cases}$$



- precise criterion:

  the "shape function"

  $$\exists w : [0,1] \to \mathbb{R}_{\geqslant 0} \;:\; \sum_{j=0}^{n-1} \left| w_{n,j} - \int_{j/n}^{(j+1)/n} w(z)\,dz \right| = O(n^{-1}) \qquad (n \to \infty).$$

- subproblem size distribution: $J_1 \overset{\mathcal{D}}{=} J_2 \overset{\mathcal{D}}{=} \text{Binomial}(n, \text{Beta}(t+1, t+1)) \pm O(1)$

  beta-binomial distribution

- binomial distribution highly concentrated $\rightsquigarrow$ $\frac{J_1}{n} \approx P \overset{\mathcal{D}}{=} \text{Beta}(t+1, t+1)$

  Chernoff etc.

  formally: $n\mathbb{P}[J_1 = \lfloor zn \rfloor] = f_{\text{Beta}}(z) \pm O(n^{-1})$

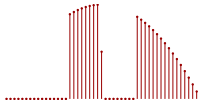
  $= z^t(1-z)^t / B(t+1, t+1)$

$\rightsquigarrow$ $w(z) = 2\left[z \in \vdash\!\!\blacksquare\!\!\dashv\right] \cdot f_{\text{Beta}}(z)$ valid shape function

Shape Function

$w_{n,j}$ *"smooth"*

$$w_{n,j} = \begin{cases} 1 \cdot \mathbb{P}[J_1 = j] & \text{if } \frac{j}{n-1} = \frac{1}{2} \\ 2 \cdot \mathbb{P}[J_1 = j] & \text{if } \frac{j}{n-1} \in \vdash\!\!\blacksquare\!\!\dashv \\ 0 & \text{otherwise} \end{cases}$$



- precise criterion:

the "shape function"

$$\exists w : [0,1] \to \mathbb{R}_{\geqslant 0} \;:\; \sum_{j=0}^{n-1} \left| w_{n,j} - \int_{j/n}^{(j+1)/n} w(z)\,dz \right| \;=\; O(n^{-1}) \qquad (n \to \infty).$$

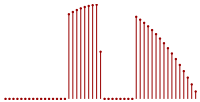
- subproblem size distribution: $J_1 \overset{\mathcal{D}}{=} J_2 \overset{\mathcal{D}}{=} \text{Binomial}(n, \text{Beta}(t+1, t+1)) \pm O(1)$ — *beta-binomial distribution*; median of $2t+1$ i.i.d. Uniform$[0,1]$ rv.
- binomial distribution highly concentrated $\rightsquigarrow$ $\frac{J_1}{n} \approx P \overset{\mathcal{D}}{=} \text{Beta}(t+1, t+1)$ — Chernoff etc.
  formally: $n\mathbb{P}\big[J_1 = \lfloor zn \rfloor\big] = f_{\text{Beta}}(z) \pm O(n^{-1})$, with $= z^t(1-z)^t / B(t+1, t+1)$

$\rightsquigarrow$ $w(z) = 2\big[z \in \vdash\!\!\blacksquare\!\!\dashv\big] \cdot f_{\text{Beta}}(z)$ valid shape function

Shape Function

$w_{n,j}$ *"smooth"*

$$w_{n,j} = \begin{cases} 1 \cdot \mathbb{P}[J_1 = j] & \text{if } \frac{j}{n-1} = \frac{1}{2} \\ 2 \cdot \mathbb{P}[J_1 = j] & \text{if } \frac{j}{n-1} \in \vdash\!\!\blacksquare\!\!\dashv \\ 0 & \text{otherwise} \end{cases}$$



- precise criterion:

$w(z) \approx n w_{n,zn}$

the "shape function"

$$\exists w : [0,1] \to \mathbb{R}_{\geqslant 0} \;:\; \sum_{j=0}^{n-1} \left| w_{n,j} - \int_{j/n}^{(j+1)/n} w(z)\,dz \right| \;=\; O(n^{-1}) \qquad (n \to \infty).$$

beta-binomial distribution

- subproblem size distribution: $J_1 \overset{\mathcal{D}}{=} J_2 \overset{\mathcal{D}}{=} \text{Binomial}(n, \text{Beta}(t+1, t+1)) \pm O(1)$

median of $2t+1$ i.i.d. $\text{Uniform}[0,1]$ rv.

Chernoff etc.

- binomial distribution highly concentrated $\rightsquigarrow$ $\frac{J_1}{n} \overset{\mathcal{D}}{\approx} P \overset{\mathcal{D}}{=} \text{Beta}(t+1, t+1)$

formally: $n\mathbb{P}\big[J_1 = \lfloor zn \rfloor\big] = f_{\text{Beta}}(z) \pm O(n^{-1})$

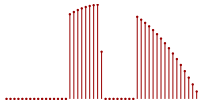
$= z^t(1-z)^t / B(t+1, t+1)$

$\rightsquigarrow$ $w(z) = 2\big[z \in \vdash\!\!\blacksquare\!\!\dashv\big] \cdot f_{\text{Beta}}(z)$ valid shape function

Shape Function

$w_{n,j}$ *"smooth"*

$$w_{n,j} = \begin{cases} 1 \cdot \mathbb{P}[J_1 = j] & \text{if } \frac{j}{n-1} = \frac{1}{2} \\ 2 \cdot \mathbb{P}[J_1 = j] & \text{if } \frac{j}{n-1} \in \vdash\!\!\blacksquare\!\!\dashv \\ 0 & \text{otherwise} \end{cases}$$



- precise criterion:

$w(z) \approx n w_{n,zn}$ … with guaranteed convergence rate

the "shape function"

$$\exists w : [0,1] \to \mathbb{R}_{\geqslant 0} \;:\; \sum_{j=0}^{n-1} \left| w_{n,j} - \int_{j/n}^{(j+1)/n} w(z)\,dz \right| \;=\; O(n^{-1}) \qquad (n \to \infty).$$

beta-binomial distribution

- subproblem size distribution: $J_1 \overset{\mathcal{D}}{=} J_2 \overset{\mathcal{D}}{=} \text{Binomial}(n, \text{Beta}(t+1, t+1)) \pm O(1)$

median of $2t+1$ i.i.d. Uniform$[0,1]$ rv.

Chernoff etc.

- binomial distribution highly concentrated $\rightsquigarrow$ $\frac{J_1}{n} \approx P \overset{\mathcal{D}}{=} \text{Beta}(t+1, t+1)$

formally: $n\mathbb{P}\big[J_1 = \lfloor zn \rfloor\big] = f_{\text{Beta}}(z) \pm O(n^{-1})$

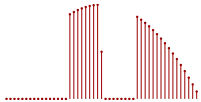
$= z^t(1-z)^t / B(t+1, t+1)$

$\rightsquigarrow$ $w(z) = 2\big[z \in \vdash\!\!\blacksquare\!\!\dashv\big] \cdot f_{\text{Beta}}(z)$ valid shape function

Shape Function

$w_{n,j}$ *"smooth"*

$$w_{n,j} = \begin{cases} 1 \cdot \mathbb{P}[J_1 = j] & \text{if } \frac{j}{n-1} = \frac{1}{2} \\ 2 \cdot \mathbb{P}[J_1 = j] & \text{if } \frac{j}{n-1} \in \vdash\!\!\blacksquare\!\!\dashv \\ 0 & \text{otherwise} \end{cases}$$



- precise criterion:

$w(z) \approx n w_{n,zn}$ … with guaranteed convergence rate

the "shape function"

$$\exists w : [0,1] \to \mathbb{R}_{\geqslant 0} \;:\; \sum_{j=0}^{n-1} \left| w_{n,j} - \int_{j/n}^{(j+1)/n} w(z)\,dz \right| \;=\; O(n^{-1}) \qquad (n \to \infty).$$

- subproblem size distribution: $J_1 \overset{\mathcal{D}}{=} J_2 \overset{\mathcal{D}}{\approx} \text{Binomial}(n, \text{Beta}(t+1, t+1)) \pm O(1)$ *beta-binomial distribution*

- binomial distribution highly concentrated ⇝ $\frac{J_1}{n} \overset{\mathcal{D}}{\approx} P \overset{\mathcal{D}}{=} \text{Beta}(t+1, t+1)$ Chernoff etc.

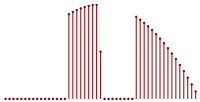
  formally: $n\mathbb{P}\big[J_1 = \lfloor zn \rfloor\big] = f_{\text{Beta}}(z) \pm O(n^{-1})$

⇝ $w(z) = 2\big[z \in \vdash\!\!\blacksquare\!\!\dashv\big] \cdot f_{\text{Beta}}(z)$ valid shape function

Shape Function

$w_{n,j}$ *"smooth"*

$$w_{n,j} = \begin{cases} 1 \cdot \mathbb{P}[J_1 = j] & \text{if } \frac{j}{n-1} = \frac{1}{2} \\ 2 \cdot \mathbb{P}[J_1 = j] & \text{if } \frac{j}{n-1} \in \vdash\!\!\blacksquare\!\!\dashv \\ 0 & \text{otherwise} \end{cases}$$



- precise criterion:

$w(z) \approx n w_{n,zn}$ … with guaranteed convergence rate

the "shape function"

$$\exists w : [0,1] \to \mathbb{R}_{\geqslant 0} \;:\; \sum_{j=0}^{n-1} \left| w_{n,j} - \int_{j/n}^{(j+1)/n} w(z)\, dz \right| \;=\; O(n^{-1}) \qquad (n \to \infty).$$

- subproblem size distribution: $J_1 \overset{\mathcal{D}}{=} J_2 \overset{\mathcal{D}}{=} \text{Binomial}(n, \text{Beta}(t+1, t+1)) \pm O(1)$ *beta-binomial distribution*

- binomial distribution highly concentrated $\rightsquigarrow$ $\frac{J_1}{n} \overset{\mathcal{D}}{\approx} P \overset{\mathcal{D}}{=} \text{Beta}(t+1, t+1)$ Chernoff etc. median of $2t+1$ i.i.d. $\text{Uniform}(0,1)$ r.v.

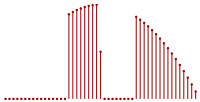
formally: $n\mathbb{P}[J_1 = \lfloor zn \rfloor] = f_{\text{Beta}}(z) \pm O(n^{-1})$ $= z^t(1-z)^t / B(t+1, t+1)$

$\rightsquigarrow$ $w(z) = 2\left[z \in \vdash\!\!\blacksquare\!\!\dashv\right] \cdot f_{\text{Beta}}(z)$ valid shape function

Shape Function

$w_{n,j}$ *"smooth"*

$$w_{n,j} = \begin{cases} 1 \cdot \mathbb{P}[J_1 = j] & \text{if } \frac{j}{n-1} = \frac{1}{2} \\ 2 \cdot \mathbb{P}[J_1 = j] & \text{if } \frac{j}{n-1} \in \vdash\!\!\blacksquare\!\!\dashv \\ 0 & \text{otherwise} \end{cases}$$



- precise criterion:

$w(z) \approx n w_{n,zn}$ … with guaranteed convergence rate

the "shape function"

$$\exists w : [0,1] \to \mathbb{R}_{\geqslant 0} \;:\; \sum_{j=0}^{n-1} \left| w_{n,j} - \int_{j/n}^{(j+1)/n} w(z)\,dz \right| \;=\; O(n^{-1}) \qquad (n \to \infty).$$

- subproblem size distribution: $J_1 \overset{\mathcal{D}}{=} J_2 \overset{\mathcal{D}}{\approx} \text{Binomial}(n, \text{Beta}(t+1, t+1)) \pm O(1)$

beta-binomial distribution

- binomial distribution highly concentrated $\rightsquigarrow$ $\frac{J_1}{n} \overset{\mathcal{D}}{\approx} P \overset{\mathcal{D}}{=} \text{Beta}(t+1, t+1)$

Chernoff etc.

median of $2t+1$ i.i.d. $\text{Uniform}(0,1)$ r.v.

formally: $n\mathbb{P}\big[J_1 = \lfloor zn \rfloor\big] = f_{\text{Beta}}(z) \pm O(n^{-1})$

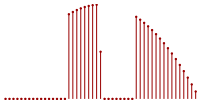
$= z^t(1-z)^t / B(t+1, t+1)$

$\rightsquigarrow$ $w(z) = 2\big[z \in \vdash\!\!\blacksquare\!\!\dashv\big] \cdot f_{\text{Beta}}(z)$ valid shape function

Shape Function

$w_{n,j}$ *"smooth"*

$$w_{n,j} = \begin{cases} 1 \cdot \mathbb{P}[J_1 = j] & \text{if } \frac{j}{n-1} = \frac{1}{2} \\ 2 \cdot \mathbb{P}[J_1 = j] & \text{if } \frac{j}{n-1} \in \vdash\!\!\blacksquare\!\!\dashv \\ 0 & \text{otherwise} \end{cases}$$



- precise criterion:

$w(z) \approx n w_{n,zn}$ … with guaranteed convergence rate

the "shape function"

$$\exists w : [0,1] \to \mathbb{R}_{\geqslant 0} \;:\; \sum_{j=0}^{n-1} \left| w_{n,j} - \int_{j/n}^{(j+1)/n} w(z)\,dz \right| \;=\; O(n^{-1}) \qquad (n \to \infty).$$

- subproblem size distribution: $J_1 \overset{\mathcal{D}}{=} J_2 \overset{\mathcal{D}}{=} \text{Binomial}(n, \text{Beta}(t+1, t+1)) \pm O(1)$

beta-binomial distribution

- binomial distribution highly concentrated $\rightsquigarrow$ $\frac{J_1}{n} \overset{\mathcal{D}}{\approx} P \overset{\mathcal{D}}{=} \text{Beta}(t+1, t+1)$
  formally: $n\mathbb{P}\big[J_1 = \lfloor zn \rfloor\big] = f_{\text{Beta}}(z) \pm O(n^{-1})$

Chernoff etc.

median of $2t+1$ i.i.d. $\text{Uniform}(0,1)$ r.v.

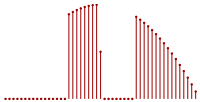
$= z^t(1-z)^t / B(t+1, t+1)$

$\rightsquigarrow$ $w(z) = 2\big[z \in \vdash\!\!\blacksquare\!\!\dashv\big] \cdot f_{\text{Beta}}(z)$ valid shape function

Shape Function

$w_{n,j}$ *"smooth"*

$$w_{n,j} = \begin{cases} 1 \cdot \mathbb{P}[J_1 = j] & \text{if } \frac{j}{n-1} = \frac{1}{2} \\ 2 \cdot \mathbb{P}[J_1 = j] & \text{if } \frac{j}{n-1} \in \vdash\!\!\blacksquare\!\!\dashv \\ 0 & \text{otherwise} \end{cases}$$



- precise criterion:

  $w(z) \approx n w_{n,zn}$ … with guaranteed convergence rate

  the "shape function"

  $$\exists w : [0,1] \to \mathbb{R}_{\geqslant 0} \;:\; \sum_{j=0}^{n-1} \left| w_{n,j} - \int_{j/n}^{(j+1)/n} w(z)\,dz \right| = O(n^{-1}) \qquad (n \to \infty).$$

- subproblem size distribution: $J_1 \overset{\mathcal{D}}{=} J_2 \overset{\mathcal{D}}{=} \text{Binomial}(n, \text{Beta}(t+1, t+1)) \pm O(1)$

  beta-binomial distribution

- binomial distribution highly concentrated $\rightsquigarrow$ $\frac{J_1}{n} \overset{\mathcal{D}}{\approx} P \overset{\mathcal{D}}{=} \text{Beta}(t+1, t+1)$

  Chernoff etc. — median of $2t+1$ i.i.d. $\text{Uniform}(0,1)$ r.v.

  formally: $n\mathbb{P}\big[J_1 = \lfloor zn \rfloor\big] = f_{\text{Beta}}(z) \pm O(n^{-1})$

  $= z^t(1-z)^t / B(t+1, t+1)$

$\rightsquigarrow$ $w(z) = 2\big[z \in \vdash\!\!\blacksquare\!\!\dashv\big] \cdot f_{\text{Beta}}(z)$ valid shape function

Shape Function

$w_{n,j}$ *"smooth"*

- precise criterion:

  the "shape function"

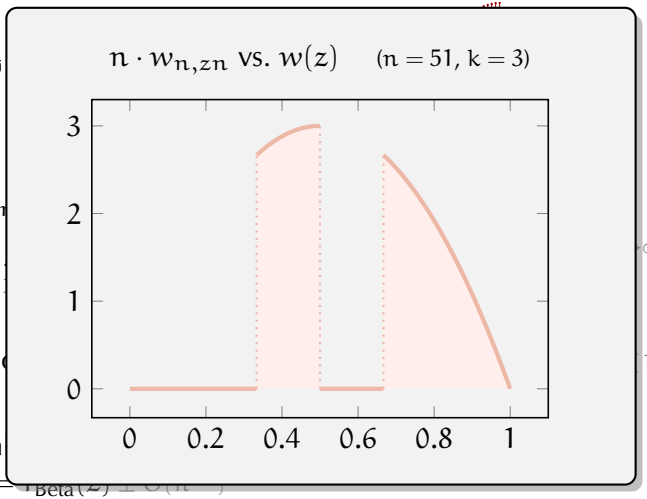
  $\exists w : [0,1] \to \mathbb{R}_{\geqslant 0}$:

- subproblem size distributi

- binomial distribution high
  formally: $n\mathbb{P}\big[J_1 = \lfloor zn \rfloor\big] = f_{\text{Beta}}(z) \pm O(n^{-\ldots})$

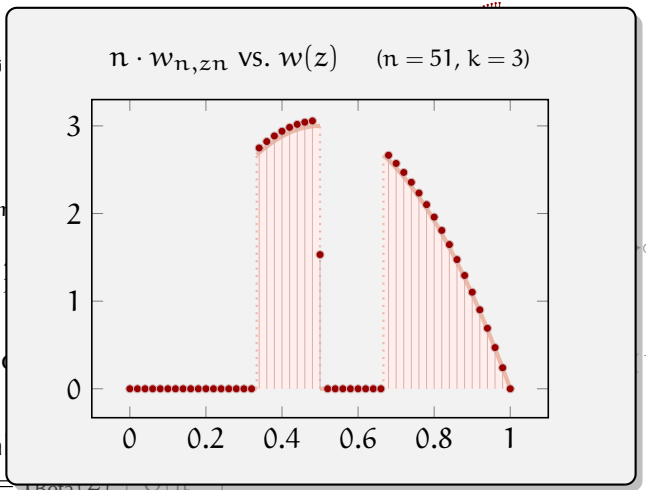
  $= z^t(1-z)^t/B(t+1,t+1)$

$\rightsquigarrow$ $w(z) = 2\big[z \in$  $\big] \cdot f_{\text{Beta}}(z)$ valid shape function



Shape Function

$w_{n,j}$ *"smooth"*

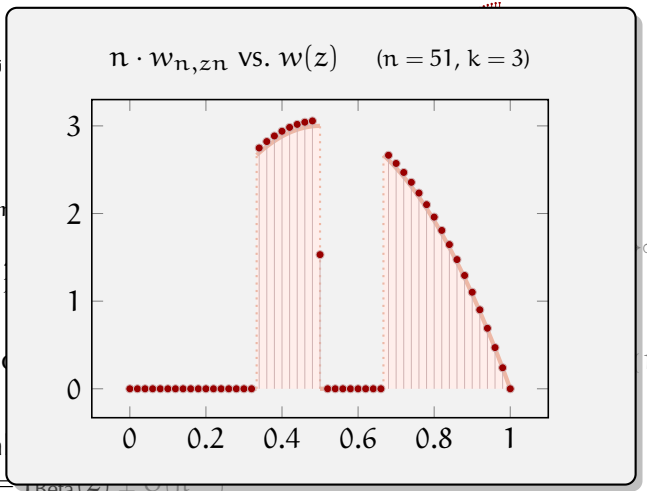


- precise criterion:
  the "shape function"
  $\exists w : [0,1] \to \mathbb{R}_{\geqslant 0}$:
- subproblem size distributi
- binomial distribution high
  formally: $n\mathbb{P}\big[J_1 = \lfloor zn \rfloor\big] =$ $f_{\text{Beta}(2)}$

$\rightsquigarrow$ $w(z) = 2\big[z \in$  $\big] \cdot f_{\text{Beta}}(z)$ valid shape function

Shape Function

$w_{n,j}$ *"smooth"*



- precise criterion:

  the "shape function"

  $\exists w : [0,1] \to \mathbb{R}_{\geqslant 0}$:

- subproblem size distributi

- binomial distribution high

  formally: $n\mathbb{P}\big[J_1 = \lfloor zn \rfloor\big] =$ $= z^t(1-z)^t/B(t+1,t+1)$

$\rightsquigarrow$ $w(z) = 2\big[z \in$ $\big] \cdot f_{\text{Beta}}(z)$ valid shape function

Expected Subproblem Size

Intuition behind shape function? $w(z) = 2\big[z \in$  $\big] \cdot f_{\mathrm{Beta}(t+1,t+1)}(z)$

- $w(z)$ is **limiting density** of **relative subproblem size** of recursive call $\left(\int_0^1 w(z)\,dz = 1\right)$
- expected fraction of elements in recursive call:

$$r := \int_0^1 z \cdot w(z)\,dz = 2\int_{z\in}$$ $$z \cdot \frac{z^t(1-z)^t}{B(t+1,t+1)}\,dz = 2\mathbb{E}\Big[\big[P \in$$ $$\big] \cdot P\Big]$$

incomplete beta integral

Expected Subproblem Size

Intuition behind shape function? $w(z) = 2\big[z \in$  $\big] \cdot f_{\text{Beta}(t+1,t+1)}(z)$

- $w(z)$ is **limiting density** of **relative subproblem size** of recursive call $\left(\int_0^1 w(z)\,dz = 1\right)$
- expected fraction of elements in recursive call:

$$r := \int_0^1 z \cdot w(z)\,dz = 2\int_{z\in$$ $$z \cdot \frac{z^t(1-z)^t}{B(t+1,t+1)}\,dz = 2\mathbb{E}\Big[\big[P \in$$ $$\big] \cdot P\Big]$$

incomplete beta integral

Expected Subproblem Size

Intuition behind shape function? $w(z) = 2\big[z \in$  $\big] \cdot f_{\text{Beta}(t+1,t+1)}(z)$

- $w(z)$ is **limiting density** of **relative subproblem size** of recursive call $\quad (\int_0^1 w(z)\,dz = 1)$
- expected fraction of elements in recursive call:

$$r := \int_0^1 z \cdot w(z)\,dz \;=\; 2\int_{z\in}$$  $$z \cdot \frac{z^t(1-z)^t}{B(t+1,t+1)}\,dz \;=\; 2\mathbb{E}\Big[\big[P \in$$ $$\big] \cdot P\Big]$$

incomplete beta integral

Expected Subproblem Size

Intuition behind shape function? $w(z) = 2\big[z \in$  $\big] \cdot f_{\text{Beta}(t+1,t+1)}(z)$

- $w(z)$ is **limiting density** of **relative subproblem size** of recursive call $\quad (\int_0^1 w(z)\,dz = 1)$
- expected fraction of elements in recursive call:

$$r := \int_0^1 z \cdot w(z)\,dz \;=\; 2\int_{z\in}\,z \cdot \frac{z^t(1-z)^t}{B(t+1,t+1)}\,dz \;=\; 2\mathbb{E}\Big[\big[P \in$$  $$\big] \cdot P\Big]$$

incomplete beta integral

sample size	fraction of elements in mergesort / recursive call	$f_{\text{Beta}(t+1,t+1)}$	$w(z)$

Expected Subproblem Size

Intuition behind shape function? $w(z) = 2\big[z \in$  $\big] \cdot f_{\mathrm{Beta}(t+1,t+1)}(z)$

- $w(z)$ is **limiting density** of **relative subproblem size** of recursive call $\quad (\int_0^1 w(z)\,dz = 1)$
- expected fraction of elements in recursive call:

$$r := \int_0^1 z \cdot w(z)\,dz \;=\; 2\int_{z\in}$$  $$z \cdot \frac{z^t(1-z)^t}{B(t+1,t+1)}\,dz \;=\; 2\mathbb{E}\Big[\big[P \in$$  $$\big] \cdot P\Big]$$

incomplete beta integral

sample size	fraction of elements in mergesort / recursive call	$f_{\mathrm{Beta}(t+1,t+1)}$	$w(z)$
none	**30.6%** mergesort | **69.4%** recursive		

Expected Subproblem Size

Intuition behind shape function? $w(z) = 2\big[z \in$  $\big] \cdot f_{\text{Beta}(t+1,t+1)}(z)$

- $w(z)$ is **limiting density** of **relative subproblem size** of recursive call $\quad (\int_0^1 w(z)\,dz = 1)$
- expected fraction of elements in recursive call:

$$r := \int_0^1 z \cdot w(z)\,dz \;=\; 2\int_{z\in}$$  $$z \cdot \frac{z^t(1-z)^t}{B(t+1,t+1)}\,dz \;=\; 2\mathbb{E}\Big[\big[P \in$$  $$\big] \cdot P\Big]$$

incomplete beta integral

sample size	fraction of elements in mergesort / recursive call	$f_{\text{Beta}(t+1,t+1)}$	$w(z)$
none	**30.6%** mergesort / **69.4%** recursive		
median of 3	**39.1%** mergesort / **60.9%** recursive		

Expected Subproblem Size

Intuition behind shape function? $w(z) = 2\big[z \in$  $\big] \cdot f_{\text{Beta}(t+1,t+1)}(z)$

- $w(z)$ is **limiting density** of **relative subproblem size** of recursive call $\quad (\int_0^1 w(z)\,dz = 1)$
- expected fraction of elements in recursive call:

$$r := \int_0^1 z \cdot w(z)\,dz \;=\; 2\int_{z\in}\; z \cdot \frac{z^t(1-z)^t}{B(t+1,t+1)}\,dz \;=\; 2\mathbb{E}\Big[\big[P \in$$  $$\big] \cdot P\Big]$$

incomplete beta integral

sample size	fraction of elements in mergesort / recursive call	$f_{\text{Beta}(t+1,t+1)}$	$w(z)$
none	**30.6%** mergesort / **69.4%** recursive		
median of 3	**39.1%** mergesort / **60.9%** recursive		
median of 5	**43.7%** mergesort / **56.3%** recursive		

Expected Subproblem Size

Intuition behind shape function? $w(z) = 2\big[z \in$  $\big] \cdot f_{\mathrm{Beta}(t+1,t+1)}(z)$

- $w(z)$ is **limiting density** of **relative subproblem size** of recursive call $\quad (\int_0^1 w(z)\,\mathrm{d}z = 1)$
- expected fraction of elements in recursive call:

$$r := \int_0^1 z \cdot w(z)\,\mathrm{d}z \;=\; 2\int_{z\in}\; z \cdot \frac{z^t(1-z)^t}{B(t+1,t+1)}\,\mathrm{d}z \;=\; 2\mathbb{E}\Big[\big[P \in$$  $$\big] \cdot P\Big]$$

incomplete beta integral

sample size	fraction of elements in mergesort / recursive call	$f_{\mathrm{Beta}(t+1,t+1)}$	$w(z)$
none	**30.6%** mergesort / **69.4% recursive**		
median of 3	**39.1%** mergesort / **60.9% recursive**		
median of 5	**43.7%** mergesort / **56.3% recursive**		
median of 7	**46.6%** mergesort / **53.4% recursive**		

Expected Subproblem Size

Intuition behind shape function? $w(z) = 2\big[z \in$  $\big] \cdot f_{\mathrm{Beta}(t+1,t+1)}(z)$

- $w(z)$ is **limiting density** of **relative subproblem size** of recursive call $\quad (\int_0^1 w(z)\,\mathrm{d}z = 1)$
- expected fraction of elements in recursive call:

$$r := \int_0^1 z \cdot w(z)\,\mathrm{d}z \;=\; 2\int_{z\in}\; z \cdot \frac{z^t(1-z)^t}{B(t+1,t+1)}\,\mathrm{d}z \;=\; 2\mathbb{E}\Big[\big[P \in$$  $\big] \cdot P\Big]$$

incomplete beta integral

sample size	fraction of elements in mergesort / recursive call	$f_{\mathrm{Beta}(t+1,t+1)}$	$w(z)$
none	**30.6%** mergesort / **69.4% recursive**		
median of 3	**39.1%** mergesort / **60.9% recursive**		
median of 5	**43.7%** mergesort / **56.3% recursive**		
median of 7	**46.6%** mergesort / **53.4% recursive**		
median of 41	**55.1%** mergesort / **44.9% recursive**		

Expected Subproblem Size

Intuition behind shape function? $w(z) = 2\big[z \in$  $\big] \cdot f_{\mathrm{Beta}(t+1,t+1)}(z)$

- $w(z)$ is **limiting density** of **relative subproblem size** of recursive call $\quad (\int_0^1 w(z)\,dz = 1)$
- expected fraction of elements in recursive call:

$$r := \int_0^1 z \cdot w(z)\,dz = 2\int_{z\in}$$  $$z \cdot \frac{z^t(1-z)^t}{B(t+1,t+1)}\,dz = 2\mathbb{E}\Big[\big[P \in$$ $$\big] \cdot P\Big]$$

incomplete beta integral

sample size	fraction of elements in mergesort / recursive call	$f_{\mathrm{Beta}(t+1,t+1)}$	$w(z)$
none	**30.6%** mergesort / **69.4% recursive**		
median of 3	**39.1%** mergesort / **60.9% recursive**		
median of 5	**43.7%** mergesort / **56.3% recursive**		
median of 7	**46.6%** mergesort / **53.4% recursive**		
median of 41	**55.1%** mergesort / **44.9% recursive**		
exact median	**50%** mergesort / **50% recursive**		

Outline

1 What is QuickXsort?

2 Recurrence

3 Toll function

4 Result

The leading term

$$\boxed{t(n) \sim K n^{\alpha} \log^{\beta} n}$$

$$t(n) = \mathbb{E}\left[\left[\frac{J_1}{n-1} \in \text{[redacted]}\right] x(J_1)\right] + \mathbb{E}\left[\left[\frac{J_2}{n-1} \in \text{[redacted]}\right] x(J_2)\right] + (n-k) + b(k)$$

Recall: $x(n) = n \lg n - 1.24n \pm O(1)$

$J_1 \overset{\mathcal{D}}{\approx} Pn$ where $P \overset{\mathcal{D}}{=} \text{Beta}(t+1, t+1)$

- $\mathbb{E}\left[\left[\frac{J_1}{n-1} \in \text{[redacted]}\right] x(J_1)\right]$

$\sim \mathbb{E}\left[\left[\ \ \in \text{[redacted]}\right] Pn \lg(Pn)\right]$

$\sim \mathbb{E}\left[\left[\ \ \in \text{[redacted]}\right] Pn \lg(\ n\ )\right] = \mathbb{E}\left[\left[P \in \text{[redacted]}\right] \cdot P\right] \cdot n \lg n$

$= \left(\mathbb{E}\left[\left[P \in \text{[redacted]}\right] \cdot (1-P)\right]\right) \cdot n \lg n$

Recall: $\gamma = 2\mathbb{E}\left[\left[P \in \text{[redacted]}\right] \cdot P\right]$

$= \frac{1}{2}(1-\gamma) \cdot n \lg n$

$\rightsquigarrow$ $c(n) \sim$ solution of $c(n) = t(n) + \int w(z) c(zn) dz$

$\gamma \cdot n \lg n = (1-\gamma) \cdot n \lg n + \gamma\gamma \cdot n \lg n$

Ansatz: $c(n) \sim \gamma \cdot n \lg n$

The leading term

$t(n) \sim Kn^{\alpha} \log^{\beta} n$

$$t(n) \;=\; \mathbb{E}\Big[\Big[\tfrac{J_1}{n-1} \in \;\Big]\, x(J_1)\Big] + \mathbb{E}\Big[\Big[\tfrac{J_2}{n-1} \in \;\Big]\, x(J_2)\Big] + (n-k) + b(k)$$

- $\mathbb{E}\Big[\Big[\tfrac{J_1}{n-1} \in \;\Big]\, x(J_1)\Big]$

Recall: $x(n) = n \lg n - 1.24n \pm O(1)$

$J_1 \overset{\mathcal{D}}{\approx} Pn$ where $P \overset{\mathcal{D}}{=} \text{Beta}(t+1, t+1)$

$$\sim \mathbb{E}\Big[\Big[\;P\;\in\;\Big]\, Pn \lg(Pn)\Big]$$

$$\sim \mathbb{E}\Big[\Big[\;P\;\in\;\Big]\, Pn \lg(\;n\;)\Big] \;=\; \mathbb{E}\Big[\Big[P \in \;\Big] \cdot P\Big] \cdot n \lg n$$

$$= \Big(\mathbb{E}\Big[\Big[P \in \;\Big] \cdot (1-P)\Big]\Big) \cdot n \lg n$$

Recall: $r = 2\mathbb{E}\Big[\Big[P \in \;\Big] \cdot P\Big]$

$$= \tfrac{1}{2}(1-r) \cdot n \lg n$$

$\rightsquigarrow$ $c(n) \sim$ solution of $c(n) = t(n) + \int w(z) c(zn)\, dz$ Ansatz: $c(n) \sim \gamma \cdot n \lg n$

$\gamma \cdot n \lg n \;=\; (1-r) \cdot n \lg n + \gamma r \cdot n \lg n$

The leading term

$t(n) \sim Kn^{\alpha}\log^{\beta} n$

$$t(n) \;=\; \mathbb{E}\Big[\Big[\tfrac{J_1}{n-1} \in \;\Big]x(J_1)\Big] + \mathbb{E}\Big[\Big[\tfrac{J_2}{n-1} \in \;\Big]x(J_2)\Big] + (n-k)+b(k)$$

- $\mathbb{E}\Big[\Big[\tfrac{J_1}{n-1} \in \;\Big]x(J_1)\Big]$

  $\sim \mathbb{E}\Big[\big[\;\; P \;\; \in \;\big]Pn\lg(Pn)\Big]$

  $\sim \mathbb{E}\Big[\big[\;\; P \;\; \in \;\big]Pn\lg(\;n\;)\Big] \;=\; \mathbb{E}\Big[\big[P \in \;\big]\cdot P\Big]\cdot n\lg n$

  $= \Big(\mathbb{E}\Big[\big[P \in \;\big]\cdot(1-P)\Big]\Big)\cdot n\lg n$

  $= \tfrac{1}{2}(1-r)\cdot n\lg n$

Recall: $x(n) = n\lg n - 1.24n \pm O(1)$
$J_1 \overset{\mathcal{D}}{\approx} Pn$ where $P \overset{\mathcal{D}}{=} \text{Beta}(t+1, t+1)$

Recall: $r = 2\mathbb{E}\big[[P \in \;]\cdot P\big]$

$\rightsquigarrow$ $c(n) \sim$ solution of $c(n) = t(n) + \int w(z)c(zn)\,dz$ Ansatz: $c(n) \sim \gamma\cdot n\lg n$

$\gamma\cdot n\lg n \;=\; (1-r)\cdot n\lg n + \gamma r\cdot n\lg n$

The leading term

$t(n) \sim Kn^{\alpha}\log^{\beta} n$

$$t(n) \;=\; \mathbb{E}\Big[\Big[\tfrac{J_1}{n-1} \in \blacksquare\Big] x(J_1)\Big] + \mathbb{E}\Big[\Big[\tfrac{J_2}{n-1} \in \blacksquare\Big] x(J_2)\Big] + (n-k) + b(k)$$

- $\mathbb{E}\Big[\Big[\tfrac{J_1}{n-1} \in \blacksquare\Big] x(J_1)\Big]$

Recall: $x(n) = n\lg n - 1.24n \pm O(1)$
$J_1 \overset{\mathcal{D}}{\approx} Pn$ where $P \overset{\mathcal{D}}{=} \mathrm{Beta}(t+1, t+1)$

$$\sim \mathbb{E}\Big[\Big[\;P\;\in \blacksquare\Big] Pn\lg(Pn)\Big]$$

$$\sim \mathbb{E}\Big[\Big[\;P\;\in \blacksquare\Big] Pn\lg(\;n\;)\Big] \;=\; \mathbb{E}\Big[\Big[P \in \blacksquare\Big] \cdot P\Big] \cdot n\lg n$$

$$= \Big(\mathbb{E}\Big[\Big[P \in \blacksquare\Big] \cdot (1-P)\Big]\Big) \cdot n\lg n$$

Recall: $r = 2\mathbb{E}\Big[\Big[P \in \blacksquare\Big] \cdot P\Big]$

$$= \tfrac{1}{2}(1-r) \cdot n\lg n$$

$\rightsquigarrow$ $c(n) \sim$ solution of $c(n) = t(n) + \int w(z)c(zn)\,dz$

Ansatz: $c(n) \sim \gamma \cdot n\lg n$

$$\gamma \cdot n\lg n \;=\; (1-r) \cdot n\lg n + \gamma r \cdot n\lg n$$

The leading term

$t(n) \sim K n^{\alpha} \log^{\beta} n$

$$t(n) \;=\; \mathbb{E}\Big[\Big[\tfrac{J_1}{n-1} \in \blacksquare\Big] x(J_1)\Big] + \mathbb{E}\Big[\Big[\tfrac{J_2}{n-1} \in \blacksquare\Big] x(J_2)\Big] + (n-k) + b(k)$$

- $\mathbb{E}\Big[\Big[\tfrac{J_1}{n-1} \in \blacksquare\Big] x(J_1)\Big]$

  Recall: $x(n) = n \lg n - 1.24n \pm O(1)$
  $J_1 \overset{\mathcal{D}}{\approx} Pn$ where $P \overset{\mathcal{D}}{=} \mathrm{Beta}(t+1, t+1)$

$$\sim \mathbb{E}\Big[\big[\;P\;\in \blacksquare\big] Pn \lg(Pn)\Big]$$

$$\sim \mathbb{E}\Big[\big[\;P\;\in \blacksquare\big] Pn \lg(\;n\;)\Big] \;=\; \mathbb{E}\Big[\big[P \in \blacksquare\big] \cdot P\Big] \cdot n \lg n$$

$$= \Big(\mathbb{E}\Big[\big[P \in \blacksquare\big] \cdot (1-P)\Big]\Big) \cdot n \lg n$$

Recall: $r = 2\mathbb{E}\Big[\big[P \in \blacksquare\big] \cdot P\Big]$

$$= \tfrac{1}{2}(1-r) \cdot n \lg n$$

$\rightsquigarrow$ $c(n) \sim$ solution of $c(n) = t(n) + \int w(z) c(zn)\,dz$ Ansatz: $c(n) \sim \gamma \cdot n \lg n$

$\gamma \cdot n \lg n \;=\; (1-r) \cdot n \lg n + \gamma r \cdot n \lg n$

The leading term

$t(n) \sim K n^{\alpha} \log^{\beta} n$

$$t(n) \;=\; \mathbb{E}\Big[\Big[\tfrac{J_1}{n-1} \in \blacksquare\Big] x(J_1)\Big] + \mathbb{E}\Big[\Big[\tfrac{J_2}{n-1} \in \blacksquare\Big] x(J_2)\Big] + (n-k) + b(k)$$

- $\mathbb{E}\Big[\Big[\tfrac{J_1}{n-1} \in \blacksquare\Big] x(J_1)\Big]$

  Recall: $x(n) = n \lg n - 1.24n \pm O(1)$
  $J_1 \overset{\mathcal{D}}{\approx} Pn$ where $P \overset{\mathcal{D}}{=} \text{Beta}(t+1, t+1)$

$$\sim \mathbb{E}\Big[\Big[\;P\;\in \blacksquare\Big] Pn \lg(Pn)\Big]$$

$$\sim \mathbb{E}\Big[\Big[\;P\;\in \blacksquare\Big] Pn \lg(\;n\;)\Big] \;=\; \mathbb{E}\Big[\Big[P \in \blacksquare\Big] \cdot P\Big] \cdot n \lg n$$

$$= \left(\mathbb{E}\Big[\Big[P \in \blacksquare\Big] \cdot (1-P)\Big]\right) \cdot n \lg n$$

Recall: $r = 2\mathbb{E}\Big[\Big[P \in \blacksquare\Big] \cdot P\Big]$

$$= \tfrac{1}{2}(1-r) \cdot n \lg n$$

$\rightsquigarrow$ $c(n) \sim$ solution of $c(n) = t(n) + \int w(z) c(zn)\, dz$

Ansatz: $c(n) \sim \gamma \cdot n \lg n$

$$\gamma \cdot n \lg n \;=\; (1-r) \cdot n \lg n + \gamma r \cdot n \lg n$$

The leading term

$$\boxed{t(n) \sim K n^{\alpha} \log^{\beta} n}$$

$$t(n) = \mathbb{E}\left[\left[\frac{J_1}{n-1} \in \right] x(J_1)\right] + \mathbb{E}\left[\left[\frac{J_2}{n-1} \in \right] x(J_2)\right] + (n-k) + b(k)$$

Recall: $x(n) = n \lg n - 1.24n \pm O(1)$

$J_1 \overset{\mathcal{D}}{\approx} Pn$ where $P \overset{\mathcal{D}}{=} \text{Beta}(t+1, t+1)$

- $$\mathbb{E}\left[\left[\frac{J_1}{n-1} \in \right] x(J_1)\right]$$

$$\sim \mathbb{E}\left[\left[\ P \ \in \right] Pn \lg(Pn)\right]$$

$$\sim \mathbb{E}\left[\left[\ P \ \in \right] Pn \lg(\ n\ )\right] = \mathbb{E}\left[\left[P \in \right] \cdot P\right] \cdot n \lg n$$

$$= \left(\mathbb{E}\left[\left[P \in \right] \cdot (1-P)\right]\right) \cdot n \lg n$$

Recall: $r = 2\mathbb{E}\left[\left[P \in \right] \cdot P\right]$

$$= \frac{1}{2}(1-r) \cdot n \lg n$$

$\rightsquigarrow$ $c(n) \sim$ solution of $c(n) = t(n) + \int w(z) c(zn) dz$

$\gamma \cdot n \lg n = (1-r) \cdot n \lg n + \gamma r \cdot n \lg n$

Ansatz: $c(n) \sim \gamma \cdot n \lg n$

The leading term

$t(n) \sim Kn^{\alpha}\log^{\beta} n$

$$t(n) \;=\; \mathbb{E}\Big[\Big[\tfrac{J_1}{n-1} \in \;\Big]x(J_1)\Big] + \mathbb{E}\Big[\Big[\tfrac{J_2}{n-1} \in \;\Big]x(J_2)\Big] + (n-k)+b(k)$$

- $\mathbb{E}\Big[\Big[\tfrac{J_1}{n-1} \in \;\Big]x(J_1)\Big]$

Recall: $x(n) = n\lg n - 1.24n \pm O(1)$
$J_1 \overset{\mathcal{D}}{\approx} Pn$ where $P \overset{\mathcal{D}}{=} \text{Beta}(t+1, t+1)$

$$\sim \mathbb{E}\Big[\big[\;P\;\in\;\big]Pn\lg(Pn)\Big]$$

$$\sim \mathbb{E}\Big[\big[\;P\;\in\;\big]Pn\lg(\;n\;)\Big] \;=\; \mathbb{E}\Big[\big[P \in \;\big]\cdot P\Big]\cdot n\lg n$$

$$= \left(\mathbb{E}\Big[\big[P \in \;\big]\cdot(1-P)\Big]\right)\cdot n\lg n$$

Recall: $r = 2\mathbb{E}\Big[\big[P \in \;\big]\cdot P\Big]$

$$= \tfrac{1}{2}(1-r)\cdot n\lg n$$

$\rightsquigarrow$ $c(n) \sim$ solution of $c(n) = t(n) + \int w(z)c(zn)\,dz$ Ansatz: $c(n) \sim \gamma\cdot n\lg n$

$\gamma\cdot n\lg n \;=\; (1-r)\cdot n\lg n + \gamma r\cdot n\lg n$

The leading term

$t(n) \sim Kn^{\alpha}\log^{\beta} n$

$$t(n) \;=\; \mathbb{E}\Big[\Big[\tfrac{J_1}{n-1} \in \;\Big]x(J_1)\Big] + \mathbb{E}\Big[\Big[\tfrac{J_2}{n-1} \in \;\Big]x(J_2)\Big] + (n-k)+b(k)$$

- $\mathbb{E}\Big[\Big[\tfrac{J_1}{n-1} \in \;\Big]x(J_1)\Big]$

Recall: $x(n) = n\lg n - 1.24n \pm O(1)$
$J_1 \overset{\mathcal{D}}{\approx} Pn$ where $P \overset{\mathcal{D}}{=} \mathrm{Beta}(t+1, t+1)$

$$\sim \mathbb{E}\Big[\Big[\;P\;\in\;\Big]Pn\lg(Pn)\Big]$$

$$\sim \mathbb{E}\Big[\Big[\;P\;\in\;\Big]Pn\lg(\;n\;)\Big] \;=\; \mathbb{E}\Big[\Big[P \in \;\Big]\cdot P\Big]\cdot n\lg n$$

$$= \left(\mathbb{E}\Big[\Big[P \in \;\Big]\cdot(1-P)\Big]\right)\cdot n\lg n$$

Recall: $r = 2\mathbb{E}\Big[\Big[P \in \;\Big]\cdot P\Big]$

$$= \tfrac{1}{2}(1-r)\cdot n\lg n$$

$\rightsquigarrow$ $c(n) \sim$ solution of $c(n) = t(n) + \int w(z)c(zn)\,dz$

Ansatz: $c(n) \sim \gamma\cdot n\lg n$

$$\gamma\cdot n\lg n \;=\; (1-r)\cdot n\lg n + \gamma r\cdot n\lg n$$

The leading term

$t(n) \sim Kn^{\alpha}\log^{\beta} n$

$$t(n) \;=\; \mathbb{E}\Big[\Big[\tfrac{J_1}{n-1} \in \;\Big]x(J_1)\Big] + \mathbb{E}\Big[\Big[\tfrac{J_2}{n-1} \in \;\Big]x(J_2)\Big] + (n-k)+b(k)$$

- $\mathbb{E}\Big[\Big[\tfrac{J_1}{n-1} \in \;\Big]x(J_1)\Big]$

Recall: $x(n) = n\lg n - 1.24n \pm O(1)$
$J_1 \overset{\mathcal{D}}{\approx} Pn$ where $P \overset{\mathcal{D}}{=} \mathrm{Beta}(t+1, t+1)$

$$\sim \mathbb{E}\Big[\big[\;P\;\in\;\big]Pn\lg(Pn)\Big]$$

$$\sim \mathbb{E}\Big[\big[\;P\;\in\;\big]Pn\lg(\;n\;)\Big] \;=\; \mathbb{E}\Big[\big[P \in \;\big]\cdot P\Big]\cdot n\lg n$$

$$= \left(\mathbb{E}\Big[\big[P \in \;\big]\cdot(1-P)\Big]\right)\cdot n\lg n$$

Recall: $r = 2\mathbb{E}\Big[\big[P \in \;\big]\cdot P\Big]$

$$= \tfrac{1}{2}(1-r)\cdot n\lg n$$

$\rightsquigarrow$ $c(n) \sim$ solution of $c(n) = t(n) + \int w(z)c(zn)\,dz$

Ansatz: $c(n) \sim \gamma\cdot n\lg n$

$$\gamma\cdot n\lg n \;=\; (1-r)\cdot n\lg n + \gamma r\cdot n\lg n$$

The leading term

$t(n) \sim Kn^{\alpha}\log^{\beta} n$

$$t(n) \;=\; \mathbb{E}\Big[\Big[\tfrac{J_1}{n-1} \in \;\Big]x(J_1)\Big] + \mathbb{E}\Big[\Big[\tfrac{J_2}{n-1} \in \;\Big]x(J_2)\Big] + (n-k)+b(k)$$

- $\mathbb{E}\Big[\Big[\tfrac{J_1}{n-1} \in \;\Big]x(J_1)\Big]$

Recall: $x(n) = n\lg n - 1.24n \pm O(1)$
$J_1 \overset{\mathcal{D}}{\approx} Pn$ where $P \overset{\mathcal{D}}{=} \text{Beta}(t+1, t+1)$

$$\sim \mathbb{E}\Big[\big[\;P\;\in\;\big]Pn\lg(Pn)\Big]$$

$$\sim \mathbb{E}\Big[\big[\;P\;\in\;\big]Pn\lg(\;n\;)\Big] \;=\; \mathbb{E}\Big[\big[P \in \;\big]\cdot P\Big]\cdot n\lg n$$

$$= \left(\mathbb{E}\Big[\big[P \in \;\big]\cdot(1-P)\Big]\right)\cdot n\lg n$$

Recall: $r = 2\mathbb{E}\Big[\big[P \in \;\big]\cdot P\Big]$

$$= \tfrac{1}{2}(1-r)\cdot n\lg n$$

$\rightsquigarrow$ $c(n) \sim$ solution of $c(n) = t(n) + \int w(z)c(zn)\,dz$

Ansatz: $c(n) \sim \gamma\cdot n\lg n$

$$\gamma\cdot n\lg n \;=\; (1-r)\cdot n\lg n + \gamma r\cdot n\lg n$$

The leading term

$$\boxed{t(n) \sim K n^{\alpha} \log^{\beta} n}$$

$$t(n) = \mathbb{E}\left[\left[\frac{J_1}{n-1} \in \right] x(J_1)\right] + \mathbb{E}\left[\left[\frac{J_2}{n-1} \in \right] x(J_2)\right] + (n-k) + b(k)$$

Recall: $x(n) = n \lg n - 1.24n \pm O(1)$

$J_1 \overset{\mathcal{D}}{\approx} Pn$ where $P \overset{\mathcal{D}}{=} \mathrm{Beta}(t+1, t+1)$

- $$\mathbb{E}\left[\left[\frac{J_1}{n-1} \in \right] x(J_1)\right]$$

$$\sim \mathbb{E}\left[\left[\ P \ \in \right] Pn \lg(Pn)\right]$$

$$\sim \mathbb{E}\left[\left[\ P \ \in \right] Pn \lg(\ n\ )\right] = \mathbb{E}\left[\left[P \in \right] \cdot P\right] \cdot n \lg n$$

$$= \left(\mathbb{E}\left[\left[P \in \right] \cdot (1-P)\right]\right) \cdot n \lg n$$

Recall: $r = 2\mathbb{E}\left[\left[P \in \right] \cdot P\right]$

$$= \frac{1}{2}(1-r) \cdot n \lg n$$

$\rightsquigarrow$ $c(n) \sim$ solution of $c(n) = t(n) + \int w(z) c(zn) dz$

Ansatz: $c(n) \sim \gamma \cdot n \lg n$

$$\gamma \cdot \cancel{n \lg n} = (1-r) \cdot \cancel{n \lg n} + \gamma r \cdot \cancel{n \lg n}$$

The leading term

$t(n) \sim Kn^{\alpha}\log^{\beta} n$

$t(n) = \mathbb{E}\Big[\Big[\frac{J_1}{n-1} \in \quad\Big] x(J_1)\Big] + \mathbb{E}\Big[\Big[\frac{J_2}{n-1} \in \quad\Big] x(J_2)\Big] + (n-k) + b(k)$

- $\mathbb{E}\Big[\Big[\frac{J_1}{n-1} \in \quad\Big] x(J_1)\Big]$

Recall: $x(n) = n\lg n - 1.24n \pm O(1)$
$J_1 \overset{\mathcal{D}}{\approx} Pn$ where $P \overset{\mathcal{D}}{=} \text{Beta}(t+1, t+1)$

$$\sim \mathbb{E}\Big[\Big[\ P\ \in \quad\Big] Pn\lg(Pn)\Big]$$

$$\sim \mathbb{E}\Big[\Big[\ P\ \in \quad\Big] Pn\lg(\ n\ )\Big] = \mathbb{E}\Big[\Big[P \in \quad\Big] \cdot P\Big] \cdot n\lg n$$

$$= \Big(\mathbb{E}\Big[\Big[P \in \quad\Big] \cdot (1-P)\Big]\Big) \cdot n\lg n$$

Recall: $r = 2\mathbb{E}\Big[\Big[P \in \quad\Big] \cdot P\Big]$

$$= \tfrac{1}{2}(1-r) \cdot n\lg n$$

$\rightsquigarrow$ $c(n) \sim$ solution of $c(n) = t(n) + \int w(z)c(zn)\,dz$

Ansatz: $c(n) \sim \gamma \cdot n\lg n$

$\gamma \cdot n\lg n = (1-r) \cdot n\lg n + \gamma r \cdot n\lg n$ $\rightsquigarrow$ $\gamma = 1$ $\quad c(n) \sim x(n)!$

The leading term

$t(n) \sim Kn^{\alpha} \log^{\beta} n$

$$t(n) = \mathbb{E}\left[\left[\tfrac{J_1}{n-1} \in \text{▬▬}\right] x(J_1)\right] + \mathbb{E}\left[\left[\tfrac{J_2}{n-1} \in \text{▬▬}\right] x(J_2)\right] + (n-k) + b(k)$$

- $\mathbb{E}\left[\left[\frac{J_1}{n-}\right.\right.$)

$\sim \mathbb{E}\left[\left[\right.\right.$ P

$\sim \mathbb{E}\left[\left[\right.\right.$ P

$\lg n$

Recall: $r = 2\mathbb{E}\left[\right.$

$\rightsquigarrow$ $c(n) \sim$ solu… $n \lg n$

$\gamma \cdot n \lg n = (1-r) \cdot n \lg n + \gamma r \cdot n \lg n$ $\rightsquigarrow$ $\gamma = 1$ $c(n) \sim x(n)!$

$\rightsquigarrow$ **QuickMergesort is optimal w.r.t. leading term!**

The leading term

$t(n) \sim Kn^{\alpha} \log^{\beta} n$

$$t(n) = \mathbb{E}\left[\left[\tfrac{J_1}{n-1} \in \text{▬▬}\right] x(J_1)\right] + \mathbb{E}\left[\left[\tfrac{J_2}{n-1} \in \text{▬▬}\right] x(J_2)\right] + (n-k) + b(k)$$

- $\mathbb{E}\left[\left[\frac{J_1}{n-}\right.\right.$
- $\sim \mathbb{E}\left[\left[\right.\right.$ P
- $\sim \mathbb{E}\left[\left[\right.\right.$ P

$\rightsquigarrow$ **QuickMergesort is optimal w.r.t. leading term!**

Is that surprising?

$\lg n$

Recall: $r = 2\mathbb{E}\left[\right.$

$\rightsquigarrow$ $c(n) \sim$ solu

$n \lg n$

$\gamma \cdot n \lg n = (1-r) \cdot n \lg n + \gamma r \cdot n \lg n$ $\rightsquigarrow$ $\gamma = 1$ $c(n) \sim x(n)!$

The leading term

$t(n) \sim K n^{\alpha} \log^{\beta} n$

$t(n) = \mathbb{E}\left[\left[\frac{J_1}{n-1} \in \right]x(J_1)\right] + \mathbb{E}\left[\left[\frac{J_2}{n-1} \in \right]x(J_2)\right] + (n-k) + b(k)$

- $\mathbb{E}\left[\left[\frac{J_1}{n-}\right.\right.$ ... $)$

$\sim \mathbb{E}\left[\left[\ P\right.\right.$

$\sim \mathbb{E}\left[\left[\ P\right.\right.$

$n \lg n$

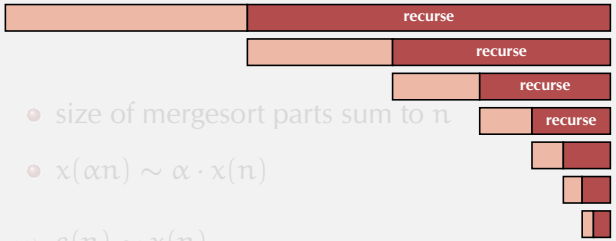
Recall: $r = 2\mathbb{E}\left[\left[\right.\right.$

$\rightsquigarrow$ $c(n) \sim$ solu... $n \lg n$

$\gamma \cdot \cancel{n \lg n} = (1-r) \cdot \cancel{n \lg n} + \gamma r \cdot \cancel{n \lg n}$ $\rightsquigarrow$ $\gamma = 1$ $c(n) \sim x(n)$!

$\rightsquigarrow$ **QuickMergesort is optimal w.r.t. leading term!**

Is that surprising?



- size of mergesort parts sum to n
- $x(\alpha n) \sim \alpha \cdot x(n)$

$\rightsquigarrow$ $c(n) \sim x(n)$

The leading term

$t(n) \sim K n^{\alpha} \log^{\beta} n$

$t(n) = \mathbb{E}\left[\left[\frac{J_1}{n-1} \in \right] x(J_1)\right] + \mathbb{E}\left[\left[\frac{J_2}{n-1} \in \right] x(J_2)\right] + (n-k) + b(k)$

- $\mathbb{E}\left[\left[\frac{J_1}{n-}\right.\right.$ ... $)$

$\sim \mathbb{E}\left[\left[\right.\right.$ P ...

$\sim \mathbb{E}\left[\left[\right.\right.$ P ...

$n \lg n$

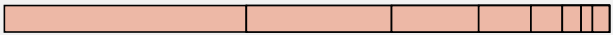
Recall: $r = 2\mathbb{E}\left[\left[\right.\right.$ ...

$\rightsquigarrow$ $c(n) \sim$ solu... $n \lg n$

$\gamma \cdot \cancel{n \lg n} = (1-r) \cdot \cancel{n \lg n} + \gamma r \cdot \cancel{n \lg n}$ $\rightsquigarrow$ $\gamma = 1$ $c(n) \sim x(n)$!

$\rightsquigarrow$ **QuickMergesort is optimal w.r.t. leading term!**

Is that surprising?



- size of mergesort parts sum to n
- $x(\alpha n) \sim \alpha \cdot x(n)$

$\rightsquigarrow$ $c(n) \sim x(n)$

The leading term

$t(n) \sim Kn^{\alpha} \log^{\beta} n$

$t(n) = \mathbb{E}\left[\left[\frac{J_1}{n-1} \in \quad\right] x(J_1)\right] + \mathbb{E}\left[\left[\frac{J_2}{n-1} \in \quad\right] x(J_2)\right] + (n-k) + b(k)$

- $\mathbb{E}\left[\left[\frac{J_1}{n-}\right.\right.$
- $\sim \mathbb{E}\left[\left[\right.\right.$ P
- $\sim \mathbb{E}\left[\left[\right.\right.$ P

$\rightsquigarrow$ **QuickMergesort is optimal w.r.t. leading term!**

Is that surprising?



- size of mergesort parts sum to n
- $x(\alpha n) \sim \alpha \cdot x(n)$ (leading term behaves linear)

$\rightsquigarrow c(n) \sim x(n)$

$\lg n$

Recall: $r = 2\mathbb{E}\left[\right.$

$\rightsquigarrow c(n) \sim$ solu

$n \lg n$

$\gamma \cdot n \lg n = (1-r) \cdot n \lg n + \gamma r \cdot n \lg n$ $\rightsquigarrow$ $\gamma = 1$ $c(n) \sim x(n)$!

The leading term

$t(n) \sim Kn^{\alpha} \log^{\beta} n$

$$t(n) = \mathbb{E}\left[\left[\tfrac{J_1}{n-1} \in \text{▬▬}\right] x(J_1)\right] + \mathbb{E}\left[\left[\tfrac{J_2}{n-1} \in \text{▬▬}\right] x(J_2)\right] + (n-k) + b(k)$$

- $\mathbb{E}\left[\left[\frac{J_1}{n-}\right.\right.$

$\sim \mathbb{E}\left[\left[\right.\right.$ P

$\sim \mathbb{E}\left[\left[\right.\right.$ P

Recall: $r = 2\mathbb{E}\left[\right.$

$\rightsquigarrow$ $c(n) \sim$ solu

$\gamma \cdot n \lg n = (1-r) \cdot n \lg n + \gamma r \cdot n \lg n$ $\rightsquigarrow$ $\gamma = 1$ $c(n) \sim x(n)$!

$\rightsquigarrow$ **QuickMergesort is optimal w.r.t. leading term!**

Is that surprising?



- size of mergesort parts sum to n
- $x(\alpha n) \sim \alpha \cdot x(n)$ (leading term behaves linear)

$\rightsquigarrow$ $c(n) \sim x(n)$

$n \lg n$

The leading term

$t(n) \sim Kn^{\alpha} \log^{\beta} n$

$t(n) = \mathbb{E}\left[\left[\frac{J_1}{n-1} \in \text{▬▬}\right] x(J_1)\right] + \mathbb{E}\left[\left[\frac{J_2}{n-1} \in \text{▬▬}\right] x(J_2)\right] + (n-k) + b(k)$

- $\mathbb{E}\left[\left[\frac{J_1}{n-}\right.\right.$
- $\sim \mathbb{E}\left[\left[\right.\right.$ P
- $\sim \mathbb{E}\left[\left[\right.\right.$ P

$\lg n$

Recall: $r = 2\mathbb{E}\left[\right.$

$\rightsquigarrow$ $c(n) \sim$ solu… $n \lg n$

$\gamma \cdot n \lg n = (1-r) \cdot n \lg n + \gamma r \cdot n \lg n$ $\rightsquigarrow$ $\gamma = 1$ $c(n) \sim x(n)$!

$\rightsquigarrow$ QuickMergesort is optimal w.r.t. leading term!

Is that surprising?



- size of mergesort parts sum to n
- $x(\alpha n) \sim \alpha \cdot x(n)$ (leading term behaves linear)

$\rightsquigarrow$ $c(n) \sim x(n)$ (that was known previously)

The leading term

$t(n) \sim Kn^{\alpha} \log^{\beta} n$

$t(n) = \mathbb{E}\left[\left[\frac{J_1}{n-1} \in \text{▬▬}\right] x(J_1)\right] + \mathbb{E}\left[\left[\frac{J_2}{n-1} \in \text{▬▬}\right] x(J_2)\right] + (n-k) + b(k)$

- $\mathbb{E}\left[\left[\frac{J_1}{n-}\right.\right.$
- $\sim \mathbb{E}\left[\left[\right.\right.$ P
- $\sim \mathbb{E}\left[\left[\right.\right.$ P

$\rightsquigarrow$ **QuickMergesort is optimal w.r.t. leading term!**

Is that surprising?



- size of mergesort parts sum to n
- $x(\alpha n) \sim \alpha \cdot x(n)$ (leading term behaves linear)

$\rightsquigarrow$ $c(n) \sim x(n)$ (that was known previously) … *algorithm details **do not matter**!*

$\lg n$

Recall: $r = 2\mathbb{E}\left[\right.$

$\rightsquigarrow$ $c(n) \sim$ solu

$n \lg n$

$\gamma \cdot n \lg n = (1-r) \cdot n \lg n + \gamma r \cdot n \lg n$ $\rightsquigarrow$ $\gamma = 1$ $c(n) \sim x(n)!$

The linear term

Linear term depends on

- pivot sampling
- partitioning method
- linear term of mergesort
- but **not** on how **small subproblems** solved!

unlike for Quicksort!

Consider $c'(n) = c(n) - n \lg n$
Can show:

The linear term

Linear term depends on

- pivot sampling
- partitioning method
- linear term of mergesort
- but **not** on how **small subproblems** solved!

unlike for Quicksort!

Consider $c'(n) = c(n) - n \lg n$
Can show:

The linear term

Linear term depends on

- pivot sampling
- partitioning method
- linear term of mergesort
- but **not** on how **small subproblems** solved!

unlike for Quicksort!

Consider $c'(n) = c(n) - n \lg n$
Can show:

The linear term

Linear term depends on

- pivot sampling
- partitioning method
- linear term of mergesort
- but **not** on how **small subproblems** solved!

unlike for Quicksort!

Consider $c'(n) = c(n) - n \lg n$
Can show:

The linear term

Linear term depends on

- pivot sampling
- partitioning method
- linear term of mergesort
- but **not** on how **small subproblems** solved!

unlike for Quicksort!

Consider $c'(n) = c(n) - n \lg n$

Can show:

The linear term

Linear term depends on

- pivot sampling
- partitioning method
- linear term of mergesort
- but **not** on how **small subproblems** solved!

unlike for Quicksort!

Consider $c'(n) = c(n) - n \lg n$
Can show:

The linear term

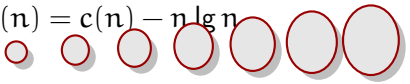
Linear term depends on

- pivot sampling
- partitioning method
- linear term of mergesort
- but **not** on how **small subproblems** solved!

unlike for Quicksort!

Consider $c'(n) = c(n) - n \lg n$

Can show:



$$
\begin{aligned}
& c'(n) \\
&= c(n) - an\lg n \\
&= \sum_{r=1}^{2} \mathbb{E}\big[A_r(J_r)c(J_r)\big] - an\lg n + t(n) \\
&= \sum_{r=1}^{2} \Big(\mathbb{E}\big[A_r(J_r)(c(J_r) - aJ_r\lg J_r)\big] - a\,\mathbb{E}\big[A_r(J_r)J_r\lg J_r\big]\Big) - an\lg n \\
&\quad + (n-1) + \mathbb{E}\big[A_2(J_2)\cdot x(J_1)\big] + \mathbb{E}\big[A_1(J_1)\cdot x(J_2)\big] \\
&= \sum_{r=1}^{2} \mathbb{E}\Big[A_r(J_r)c'(J_r)\Big] + (n-1) - an\lg n \\
&\quad + a\,\mathbb{E}\Big[\big(A_1(J_1)+A_2(J_2)\big)J_1\lg J_1\Big] + b\,\mathbb{E}[A_2(J_2)J_1] \\
&\quad + a\,\mathbb{E}\Big[\big(A_2(J_2)+A_1(J_1)\big)J_2\lg J_2\Big] + b\,\mathbb{E}[A_1(J_1)J_2] \;\pm\; O(n^{1-\varepsilon})
\end{aligned}
$$

Since $J_1 \triangleq J_2$ we can simplify

$$
\begin{aligned}
& \mathbb{E}\Big[\big(A_1(J_1)+A_2(J_2)\big)J_1\lg J_1\Big] + \mathbb{E}\Big[\big(A_2(J_2)+A_1(J_1)\big)J_2\lg J_2\Big] \\
&= \mathbb{E}\Big[\big(A_1(J_1)+A_2(J_2)\big)J_1\lg J_1\Big] + \mathbb{E}\Big[\big(A_2(J_1)+A_1(J_2)\big)J_1\lg J_1\Big] \\
&= \mathbb{E}\Big[J_1\lg J_1\cdot\big(\big(A_1(J_1)+A_1(J_2)\big)+\big(A_2(J_1)+A_2(J_2)\big)\big)\Big] \\
&= 2\mathbb{E}[J\lg J] \\
&= 2\,\mathbb{E}[\tfrac{J}{n}]\cdot n\lg n + 2\cdot\tfrac{1}{\ln 2}\mathbb{E}[\tfrac{J}{n}\ln\tfrac{J}{n}]\cdot n \\
&= n\lg n - \tfrac{1}{\ln 2}\big(H_{k+1}-H_{t+1}\big)n \;\pm\; O(n^{1-\varepsilon}).
\end{aligned}
$$

Plugging this back into our equation for $c'(n)$, we find

$$
\begin{aligned}
c'(n) &= \sum_{r=1}^{2} \mathbb{E}\Big[A_r(J_r)c'(J_r)\Big] + (n-1) - an\lg n \\
&\quad + a\Big(n\lg n - \tfrac{1}{\ln 2}\big(H_{k+1}-H_{t+1}\big)n\Big) \\
&\quad + b\Big(I_{0,\frac{1}{3}}(t+2,t+1) + I_{\frac{1}{3},\frac{2}{3}}(t+2,t+1)\Big)\cdot n \;\pm\; O(n^{1-\varepsilon}) \\
&= \sum_{r=1}^{2} \mathbb{E}\Big[A_r(J_r)c'(J_r)\Big] + t'(n)
\end{aligned}
$$

where

$$
\begin{aligned}
t'(n) &= b'n \pm O(n^{1-\varepsilon}) \\
b' &= 1 - \tfrac{a}{\ln 2}\big(H_{k+1}-H_{t+1}\big) + b\cdot H
\end{aligned}
$$

The linear term

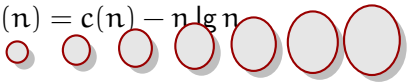
Linear term depends on

- pivot sampling
- partitioning method
- linear term of mergesort
- but **not** on how **small subproblems** solved!

unlike for Quicksort!

Consider $c'(n) = c(n) - n \lg n$

Can show:



$$c'(n) = t'(n) + \sum_{j=0}^{n-1} w_{n,j} \cdot c'(j)$$

$$\begin{aligned}
& c'(n) \\
&= c(n) - an \lg n \\
&= \sum_{r=1}^{2} \mathbb{E}\big[A_r(J_r)c(J_r)\big] - an \lg n + t(n) \\
&= \sum_{r=1}^{2} \Big(\mathbb{E}\big[A_r(J_r)(c(J_r) - aJ_r \lg J_r)\big] + a\,\mathbb{E}\big[A_r(J_r)J_r \lg J_r\big] \Big) - an \lg n \\
&\quad + (n-1) + \mathbb{E}\big[A_2(J_2) \cdot x(J_1)\big] + \mathbb{E}\big[A_1(J_1) \cdot x(J_2)\big] \\
&= \sum_{r=1}^{2} \mathbb{E}\big[A_r(J_r)c'(J_r)\big] + (n-1) - an \lg n \\
&\quad + a\,\mathbb{E}\Big[\big(A_1(J_1) + A_2(J_2)\big)J_1 \lg J_1\Big] + b\,\mathbb{E}[A_2(J_2)J_1] \\
&\quad + a\,\mathbb{E}\Big[\big(A_2(J_2) + A_1(J_1)\big)J_2 \lg J_2\Big] + b\,\mathbb{E}[A_1(J_1)J_2] \;\pm\; O(n^{1-\varepsilon})
\end{aligned}$$

Since $J_1 \triangleq J_2$ we can simplify

$$\begin{aligned}
& \mathbb{E}\Big[\big(A_1(J_1) + A_2(J_2)\big)J_1 \lg J_1\Big] + \mathbb{E}\Big[\big(A_2(J_2) + A_1(J_1)\big)J_2 \lg J_2\Big] \\
&= \mathbb{E}\Big[\big(A_1(J_1) + A_2(J_2)\big)J_1 \lg J_1\Big] + \mathbb{E}\Big[\big(A_2(J_1) + A_1(J_2)\big)J_1 \lg J_1\Big] \\
&= \mathbb{E}\Big[J_1 \lg J_1 \cdot \big(\big(A_1(J_1) + A_1(J_2)\big) + \big(A_2(J_1) + A_2(J_2)\big)\big)\Big] \\
&= 2\mathbb{E}[J \lg J] \\
&= 2\,\mathbb{E}[\tfrac{J}{n}] \cdot n \lg n + 2 \cdot \tfrac{1}{\ln 2}\mathbb{E}[\tfrac{J}{n} \ln \tfrac{J}{n}] \cdot n \\
&= n \lg n - \tfrac{1}{\ln 2}(H_{k+1} - H_{t+1})n \;\pm\; O(n^{1-\varepsilon}).
\end{aligned}$$

Plugging this back into our equation for $c'(n)$, we find

$$\begin{aligned}
c'(n) &= \sum_{r=1}^{2} \mathbb{E}\big[A_r(J_r)c'(J_r)\big] + (n-1) - an \lg n \\
&\quad + a\Big(n \lg n - \tfrac{1}{\ln 2}(H_{k+1} - H_{t+1})n\Big) \\
&\quad + b\Big(I_{0,\frac{1}{3}}(t+2, t+1) + I_{\frac{1}{3},\frac{2}{3}}(t+2, t+1)\Big) \cdot n \;\pm\; O(n^{1-\varepsilon}) \\
&= \sum_{r=1}^{2} \mathbb{E}\big[A_r(J_r)c'(J_r)\big] + t'(n)
\end{aligned}$$

where

$$\begin{aligned}
t'(n) &= b'n \pm O(n^{1-\varepsilon}) \\
b' &= 1 - \tfrac{a}{\ln 2}(H_{k+1} - H_{t+1}) + b \cdot H
\end{aligned}$$

The linear term

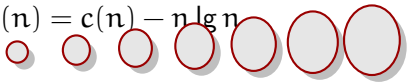
Linear term depends on

- pivot sampling
- partitioning method
- linear term of mergesort
- but **not** on how **small subproblems** solved!

unlike for Quicksort!

Consider $c'(n) = c(n) - n \lg n$

Can show:



same shape as $c(n)$

$$c'(n) \;=\; t'(n) + \sum_{j=0}^{n-1} w_{n,j} \cdot c'(j)$$

$$\begin{aligned}
& c'(n) \\
&= c(n) - an\lg n \\
&= \sum_{r=1}^{2} \mathbb{E}\big[A_r(J_r)c(J_r)\big] - an\lg n + t(n) \\
&= \sum_{r=1}^{2} \Big(\mathbb{E}\big[A_r(J_r)(c(J_r) - aJ_r\lg J_r)\big] + a\,\mathbb{E}\big[A_r(J_r)J_r\lg J_r\big] \Big) - an\lg n \\
&\quad + (n-1) + \mathbb{E}\big[A_2(J_2)\cdot x(J_1)\big] + \mathbb{E}\big[A_1(J_1)\cdot x(J_2)\big] \\
&= \sum_{r=1}^{2} \mathbb{E}\big[A_r(J_r)c'(J_r)\big] + (n-1) - an\lg n \\
&\quad + a\,\mathbb{E}\Big[\big(A_1(J_1)+A_2(J_2)\big)J_1\lg J_1\Big] + b\,\mathbb{E}[A_2(J_2)J_1] \\
&\quad + a\,\mathbb{E}\Big[\big(A_2(J_2)+A_1(J_1)\big)J_2\lg J_2\Big] + b\,\mathbb{E}[A_1(J_1)J_2] \;\pm\; O(n^{1-\varepsilon})
\end{aligned}$$

Since $J_1 \triangleq J_2$ we can simplify

$$\begin{aligned}
& \mathbb{E}\Big[\big(A_1(J_1)+A_2(J_2)\big)J_1\lg J_1\Big] + \mathbb{E}\Big[\big(A_2(J_2)+A_1(J_1)\big)J_2\lg J_2\Big] \\
&= \mathbb{E}\Big[\big(A_1(J_1)+A_2(J_2)\big)J_1\lg J_1\Big] + \mathbb{E}\Big[\big(A_2(J_1)+A_1(J_2)\big)J_1\lg J_1\Big] \\
&= \mathbb{E}\Big[J_1\lg J_1 \cdot \big((A_1(J_1)+A_1(J_2)) + (A_2(J_1)+A_2(J_2))\big)\Big] \\
&= 2\mathbb{E}[J\lg J] \\
&= 2\mathbb{E}[\tfrac{J}{n}]\cdot n\lg n + 2\cdot\tfrac{1}{\ln 2}\mathbb{E}[\tfrac{J}{n}\ln\tfrac{J}{n}]\cdot n \\
&= n\lg n - \tfrac{1}{\ln 2}(H_{k+1}-H_{t+1})n \;\pm\; O(n^{1-\varepsilon}).
\end{aligned}$$

Plugging this back into our equation for $c'(n)$, we find

$$\begin{aligned}
c'(n) &= \sum_{r=1}^{2} \mathbb{E}\big[A_r(J_r)c'(J_r)\big] + (n-1) - an\lg n \\
&\quad + a\Big(n\lg n - \tfrac{1}{\ln 2}(H_{k+1}-H_{t+1})n\Big) \\
&\quad + b\Big(I_{0,\frac{1}{3}}(t+2,t+1) + I_{\frac{1}{3},\frac{2}{3}}(t+2,t+1)\Big)\cdot n \;\pm\; O(n^{1-\varepsilon}) \\
&= \sum_{r=1}^{2} \mathbb{E}\big[A_r(J_r)c'(J_r)\big] + t'(n)
\end{aligned}$$

where

$$\begin{aligned}
t'(n) &= b'n \pm O(n^{1-\varepsilon}) \\
b' &= 1 - \tfrac{a}{\ln 2}(H_{k+1}-H_{t+1}) + b\cdot H
\end{aligned}$$

The linear term

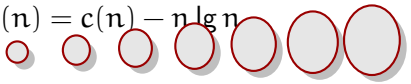
Linear term depends on

- pivot sampling
- partitioning method
- linear term of mergesort
- but **not** on how **small subproblems** solved!

unlike for Quicksort!

Consider $c'(n) = c(n) - n \lg n$

Can show:



$$c'(n) = t'(n) + \sum_{j=0}^{n-1} w_{n,j} \cdot c'(j)$$

same shape as $c(n)$

different toll function

$$\begin{aligned}
& c'(n) \\
&= c(n) - an\lg n \\
&= \sum_{r=1}^{2} \mathbb{E}\big[A_r(J_r)c(J_r)\big] - an\lg n + t(n) \\
&= \sum_{r=1}^{2} \Big(\mathbb{E}\big[A_r(J_r)(c(J_r) - aJ_r\lg J_r)\big] + a\,\mathbb{E}\big[A_r(J_r)J_r\lg J_r\big]\Big) - an\lg n \\
&\quad + (n-1) + \mathbb{E}\big[A_2(J_2)\cdot x(J_1)\big] + \mathbb{E}\big[A_1(J_1)\cdot x(J_2)\big] \\
&= \sum_{r=1}^{2} \mathbb{E}\big[A_r(J_r)c'(J_r)\big] + (n-1) - an\lg n \\
&\quad + a\,\mathbb{E}\Big[\big(A_1(J_1)+A_2(J_2)\big)J_1\lg J_1\Big] + b\,\mathbb{E}[A_2(J_2)J_1] \\
&\quad + a\,\mathbb{E}\Big[\big(A_2(J_2)+A_1(J_1)\big)J_2\lg J_2\Big] + b\,\mathbb{E}[A_1(J_1)J_2] \;\pm\; O(n^{1-\varepsilon})
\end{aligned}$$

Since $J_1 \triangleq J_2$ we can simplify

$$\begin{aligned}
& \mathbb{E}\Big[\big(A_1(J_1)+A_2(J_2)\big)J_1\lg J_1\Big] + \mathbb{E}\Big[\big(A_2(J_2)+A_1(J_1)\big)J_2\lg J_2\Big] \\
&= \mathbb{E}\Big[\big(A_1(J_1)+A_2(J_2)\big)J_1\lg J_1\Big] + \mathbb{E}\Big[\big(A_2(J_1)+A_1(J_2)\big)J_1\lg J_1\Big] \\
&= \mathbb{E}\Big[J_1\lg J_1 \cdot \big((A_1(J_1)+A_1(J_2)) + (A_2(J_1)+A_2(J_2))\big)\Big] \\
&= 2\mathbb{E}[J\lg J] \\
&= 2\mathbb{E}[\tfrac{J}{n}]\cdot n\lg n + 2\cdot\tfrac{1}{\ln 2}\mathbb{E}[\tfrac{J}{n}\ln\tfrac{J}{n}]\cdot n \\
&= n\lg n - \tfrac{1}{\ln 2}(H_{k+1} - H_{t+1})n \;\pm\; O(n^{1-\varepsilon}).
\end{aligned}$$

Plugging this back into our equation for $c'(n)$, we find

$$\begin{aligned}
c'(n) &= \sum_{r=1}^{2} \mathbb{E}\big[A_r(J_r)c'(J_r)\big] + (n-1) - an\lg n \\
&\quad + a\Big(n\lg n - \tfrac{1}{\ln 2}(H_{k+1} - H_{t+1})n\Big) \\
&\quad + b\Big(I_{0,\frac{1}{3}}(t+2,t+1) + I_{\frac{1}{3},\frac{2}{3}}(t+2,t+1)\Big)\cdot n \;\pm\; O(n^{1-\varepsilon}) \\
&= \sum_{r=1}^{2} \mathbb{E}\big[A_r(J_r)c'(J_r)\big] + t'(n)
\end{aligned}$$

where

$$\begin{aligned}
t'(n) &= b'n \pm O(n^{1-\varepsilon}) \\
b' &= 1 - \tfrac{a}{\ln 2}(H_{k+1} - H_{t+1}) + b\cdot H
\end{aligned}$$

The linear term

Linear term depends on

- pivot sampling
- partitioning method
- linear term of mergesort
- but **not** on how **small subproblems** solved!

unlike for Quicksort!

Consider $c'(n) = c(n) - n \lg n$

Can show:



same shape as $c(n)$

$$c'(n) = t'(n) + \sum_{j=0}^{n-1} w_{n,j} \cdot c'(j)$$

different toll function

$$t'(n) \sim \Big(1 - \tfrac{1}{\ln 2}\big(H_{k+1} - H_{t+1}\big) + b(1-r)\Big) \cdot n$$

$= 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{k+1}$

linear coeff. of mergesort ≈ -1.24

$$\begin{aligned}
& c'(n) \\
&= c(n) - an\lg n \\
&= \sum_{r=1}^{2} \mathbb{E}\big[A_r(J_r)c(J_r)\big] - an\lg n + t(n) \\
&= \sum_{r=1}^{2} \Big(\mathbb{E}\big[A_r(J_r)(c(J_r) - aJ_r\lg J_r)\big] + a\,\mathbb{E}\big[A_r(J_r)J_r\lg J_r\big]\Big) - an\lg n \\
&\quad + (n-1) + \mathbb{E}\big[A_2(J_2)\cdot x(J_1)\big] + \mathbb{E}\big[A_1(J_1)\cdot x(J_2)\big] \\
&= \sum_{r=1}^{2} \mathbb{E}\big[A_r(J_r)c'(J_r)\big] + (n-1) - an\lg n \\
&\quad + a\,\mathbb{E}\Big[\big(A_1(J_1) + A_2(J_2)\big)J_1\lg J_1\Big] + b\,\mathbb{E}[A_2(J_2)J_1] \\
&\quad + a\,\mathbb{E}\Big[\big(A_2(J_2) + A_1(J_1)\big)J_2\lg J_2\Big] + b\,\mathbb{E}[A_1(J_1)J_2] \;\pm\; O(n^{1-\varepsilon})
\end{aligned}$$

Since $J_1 \triangleq J_2$ we can simplify

$$\begin{aligned}
& \mathbb{E}\Big[\big(A_1(J_1) + A_2(J_2)\big)J_1\lg J_1\Big] + \mathbb{E}\Big[\big(A_2(J_2) + A_1(J_1)\big)J_2\lg J_2\Big] \\
&= \mathbb{E}\Big[\big(A_1(J_1) + A_2(J_2)\big)J_1\lg J_1\Big] + \mathbb{E}\Big[\big(A_2(J_1) + A_1(J_2)\big)J_1\lg J_1\Big] \\
&= \mathbb{E}\Big[J_1\lg J_1 \cdot \big((A_1(J_1) + A_1(J_2)) + (A_2(J_1) + A_2(J_2))\big)\Big] \\
&= 2\mathbb{E}[J\lg J] \\
&= 2\mathbb{E}[\tfrac{J}{n}]\cdot n\lg n + 2\cdot\tfrac{1}{\ln 2}\mathbb{E}[\tfrac{J}{n}\ln\tfrac{J}{n}]\cdot n \\
&= n\lg n - \tfrac{1}{\ln 2}(H_{k+1} - H_{t+1})n \;\pm\; O(n^{1-\varepsilon}).
\end{aligned}$$

Plugging this back into our equation for $c'(n)$, we find

$$\begin{aligned}
c'(n) &= \sum_{r=1}^{2} \mathbb{E}\big[A_r(J_r)c'(J_r)\big] + (n-1) - an\lg n \\
&\quad + a\Big(n\lg n - \tfrac{1}{\ln 2}(H_{k+1} - H_{t+1})n\Big) \\
&\quad + b\Big(I_{0,\frac{1}{3}}(t+2, t+1) + I_{\frac{1}{2},\frac{2}{3}}(t+2, t+1)\Big)\cdot n \;\pm\; O(n^{1-\varepsilon}) \\
&= \sum_{r=1}^{2} \mathbb{E}\big[A_r(J_r)c'(J_r)\big] + t'(n)
\end{aligned}$$

where

$$\begin{aligned}
t'(n) &= b'n \pm O(n^{1-\varepsilon}) \\
b' &= 1 - \tfrac{a}{\ln 2}(H_{k+1} - H_{t+1}) + b\cdot H
\end{aligned}$$

The linear term

Linear term depends on

- pivot sampling
- partitioning method
- linear term of mergesort
- but **not** on how **small subproblems** solved!

unlike for Quicksort!

Consider $c'(n) = c(n) - n \lg n$
Can show:

same shape as $c(n)$

$$c'(n) = t'(n) + \sum_{j=0}^{n-1} w_{n,j} \cdot c'(j)$$

different toll function

$$t'(n) \sim \left(1 - \tfrac{1}{\ln 2}\left(H_{k+1} - H_{t+1}\right) + b(1-r)\right) \cdot n$$

$= 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{k+1}$

linear coeff. of mergesort ≈ -1.24

Use *continuous master theorem* for $c'(n)$:

- $w_{n,j}$ smooth
- $t(n) \sim Kn^{\alpha} \log^{\beta} n$
- $c'(n)$ not dominated by leaves of recursion

The linear term

Linear term depends on

- pivot sampling
- partitioning method
- linear term of mergesort
- but **not** on how **small subproblems** solved!

unlike for Quicksort!

Consider $c'(n) = c(n) - n \lg n$
Can show:

same shape as $c(n)$

$$c'(n) = t'(n) + \sum_{j=0}^{n-1} w_{n,j} \cdot c'(j)$$

different toll function

$$t'(n) \sim \left(1 - \tfrac{1}{\ln 2}\left(H_{k+1} - H_{t+1}\right) + b(1-r)\right) \cdot n$$

$= 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{k+1}$

linear coeff. of mergesort ≈ -1.24

Use *continuous master theorem* for $c'(n)$:

- $w_{n,j}$ smooth
- $t(n) \sim Kn^{\alpha} \log^{\beta} n$
- $c'(n)$ not dominated by leaves of recursion

The linear term

Linear term depends on

- pivot sampling
- partitioning method
- linear term of mergesort
- but **not** on how **small subproblems** solved!

unlike for Quicksort!

Consider $c'(n) = c(n) - n \lg n$
Can show:

same shape as $c(n)$

$$c'(n) = t'(n) + \sum_{j=0}^{n-1} w_{n,j} \cdot c'(j)$$

different toll function

$$t'(n) \sim \left(1 - \tfrac{1}{\ln 2}\left(H_{k+1} - H_{t+1}\right) + b(1-r)\right) \cdot n$$

$= 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{k+1}$

linear coeff. of mergesort ≈ -1.24

Use *continuous master theorem* for $c'(n)$:

- $w_{n,j}$ smooth ✓
- $t(n) \sim Kn^{\alpha} \log^{\beta} n$
- $c'(n)$ not dominated by leaves of recursion

The linear term

Linear term depends on

- pivot sampling
- partitioning method
- linear term of mergesort
- but **not** on how **small subproblems** solved!

unlike for Quicksort!

Consider $c'(n) = c(n) - n \lg n$
Can show:

same shape as $c(n)$

$$c'(n) = t'(n) + \sum_{j=0}^{n-1} w_{n,j} \cdot c'(j)$$

different toll function

$$t'(n) \sim \left(1 - \tfrac{1}{\ln 2}\left(H_{k+1} - H_{t+1}\right) + b(1-r)\right) \cdot n$$

$= 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{k+1}$

linear coeff. of mergesort ≈ -1.24

Use *continuous master theorem* for $c'(n)$:

- $w_{n,j}$ smooth ✓
- $t(n) \sim Kn^{\alpha} \log^{\beta} n$
- $c'(n)$ not dominated by leaves of recursion

The linear term

Linear term depends on

- pivot sampling
- partitioning method
- linear term of mergesort
- but **not** on how **small subproblems** solved!

unlike for Quicksort!

Consider $c'(n) = c(n) - n \lg n$
Can show:

same shape as $c(n)$

$$c'(n) = t'(n) + \sum_{j=0}^{n-1} w_{n,j} \cdot c'(j)$$

different toll function

$$t'(n) \sim \left(1 - \tfrac{1}{\ln 2}\left(H_{k+1} - H_{t+1}\right) + b(1-r)\right) \cdot n$$

$= 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{k+1}$

linear coeff. of mergesort ≈ -1.24

Use *continuous master theorem* for $c'(n)$:

- $w_{n,j}$ smooth ✓
- $t(n) \sim Kn^{\alpha} \log^{\beta} n$ ✓
- $c'(n)$ not dominated by leaves of recursion

The linear term

Linear term depends on

- pivot sampling
- partitioning method
- linear term of mergesort
- but **not** on how **small subproblems** solved!

unlike for Quicksort!

Consider $c'(n) = c(n) - n \lg n$
Can show:

same shape as $c(n)$

$$c'(n) = t'(n) + \sum_{j=0}^{n-1} w_{n,j} \cdot c'(j)$$

different toll function

$$t'(n) \sim \left(1 - \tfrac{1}{\ln 2}\left(H_{k+1} - H_{t+1}\right) + b(1-r)\right) \cdot n$$

$= 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{k+1}$

linear coeff. of mergesort ≈ -1.24

Use *continuous master theorem* for $c'(n)$:

- $w_{n,j}$ smooth ✓
- $t(n) \sim K n^{\alpha} \log^{\beta} n$ ✓
- $c'(n)$ not dominated by leaves of recursion

The linear term

Linear term depends on

- pivot sampling
- partitioning method
- linear term of mergesort
- but **not** on how **small subproblems** solved!

unlike for Quicksort!

Consider $c'(n) = c(n) - n \lg n$
Can show:

same shape as $c(n)$

$$c'(n) = t'(n) + \sum_{j=0}^{n-1} w_{n,j} \cdot c'(j)$$

different toll function

$$t'(n) \sim \left(1 - \tfrac{1}{\ln 2}\left(H_{k+1} - H_{t+1}\right) + b(1-r)\right) \cdot n$$

$= 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{k+1}$

linear coeff. of mergesort ≈ -1.24

Use *continuous master theorem* for $c'(n)$:

- $w_{n,j}$ smooth ✓
- $t(n) \sim K n^{\alpha} \log^{\beta} n$ ✓
- $c'(n)$ not dominated by leaves of recursion ✓

The linear term

Linear term depends on

- pivot sampling
- partitioning method
- linear term of mergesort
- but **not** on how **small subproblems** solved!

unlike for Quicksort!

Consider $c'(n) = c(n) - n \lg n$
Can show:

same shape as $c(n)$

$$c'(n) = t'(n) + \sum_{j=0}^{n-1} w_{n,j} \cdot c'(j)$$

different toll function

$$t'(n) \sim \left(1 - \tfrac{1}{\ln 2}\left(H_{k+1} - H_{t+1}\right) + b(1-r)\right) \cdot n$$

$= 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{k+1}$

linear coeff. of mergesort ≈ -1.24

Use *continuous master theorem* for $c'(n)$:

- $w_{n,j}$ smooth ✓
- $t(n) \sim K n^{\alpha} \log^{\beta} n$ ✓
- $c'(n)$ not dominated by leaves of recursion ✓

$$\rightsquigarrow \quad c'(n) \sim \frac{t'(n)}{1-r}$$

Outline

1 What is QuickXsort?

2 Recurrence

3 Toll function

4 Result

Result

Comparisons in QuickMergesort:

$$c(n) \;=\; x(n) + q_t \cdot n \;\pm\; O(\log n)$$

Result

Comparisons in QuickMergesort:

$$c(n) \;=\; x(n) \;+\; q_t \cdot n \;\pm\; O(\log n)$$

Result

Comparisons in QuickMergesort:

same as X ...

$$c(n) \;=\; x(n) \;+\; q_t \cdot n \;\pm\; O(\log n)$$

Result

Comparisons in QuickMergesort:

same as X ...

$$c(n) \;=\; x(n) \;+\; q_t \cdot n \;\pm\; O(\log n)$$

... + "QuickXsort penalty"

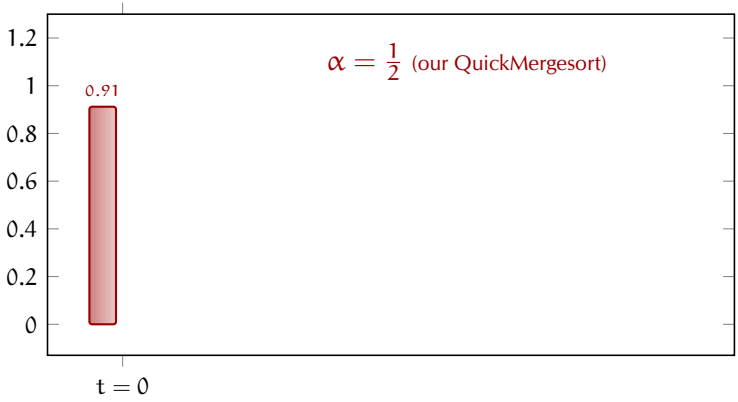
Result

Comparisons in QuickMergesort:

same as X …

$$c(n) \;=\; x(n) \;+\; q_t \cdot n \;\pm\; O(\log n)$$

… + “QuickXsort penalty”



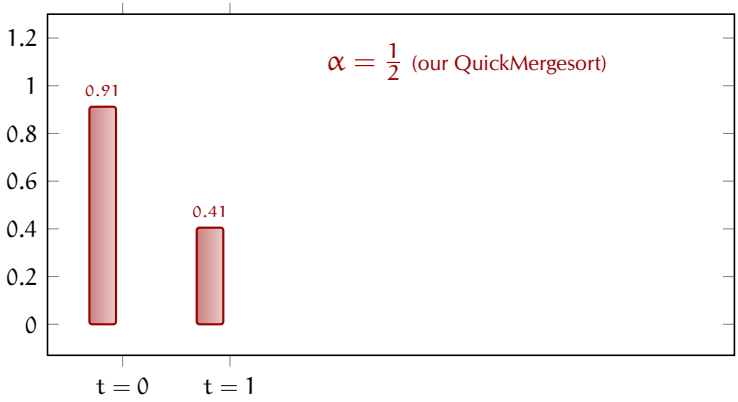
Result

Comparisons in QuickMergesort:

same as X …

$$c(n) = x(n) + q_t \cdot n \pm O(\log n)$$

… + "QuickXsort penalty"



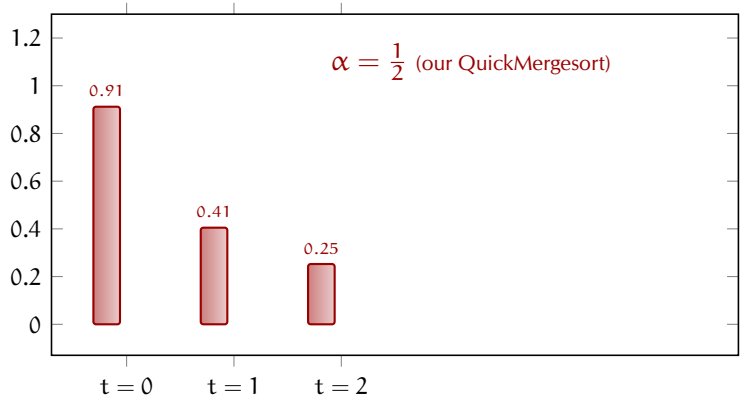
Result

Comparisons in QuickMergesort:

same as X ...

$$c(n) = x(n) + q_t \cdot n \pm O(\log n)$$

... + "QuickXsort penalty"



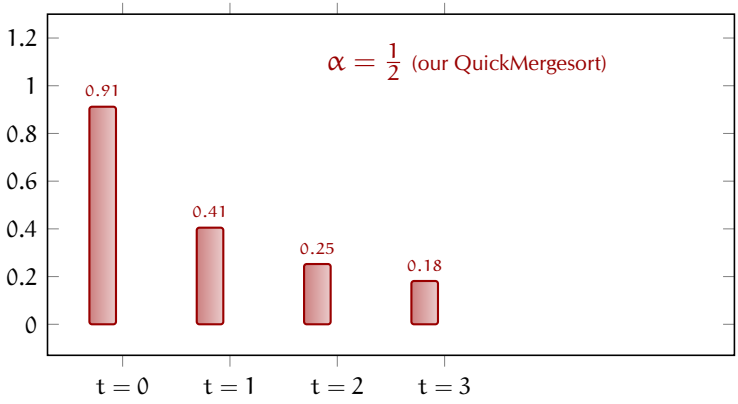
Result

Comparisons in QuickMergesort:

same as X …

$$c(n) = x(n) + q_t \cdot n \pm O(\log n)$$

… + "QuickXsort penalty"



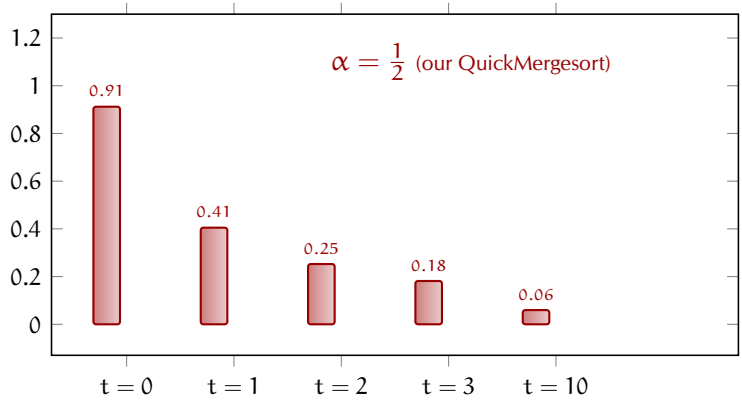
Result

Comparisons in QuickMergesort:

same as X ...

$$c(n) = x(n) + q_t \cdot n \pm O(\log n)$$

... + "QuickXsort penalty"



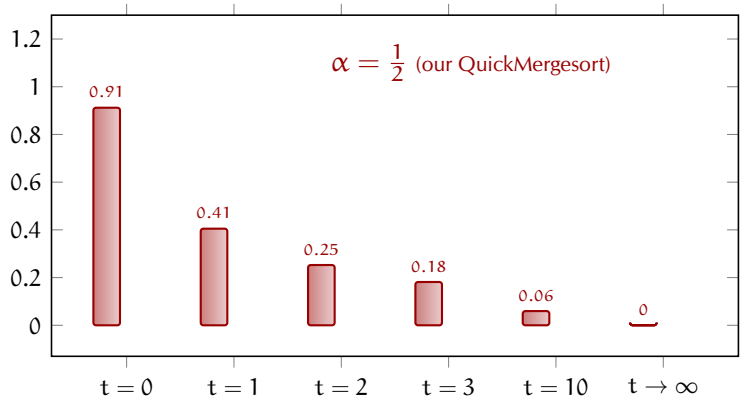
Result

Comparisons in QuickMergesort:

same as X ...

$$c(n) = x(n) + q_t \cdot n \pm O(\log n)$$

... + "QuickXsort penalty"



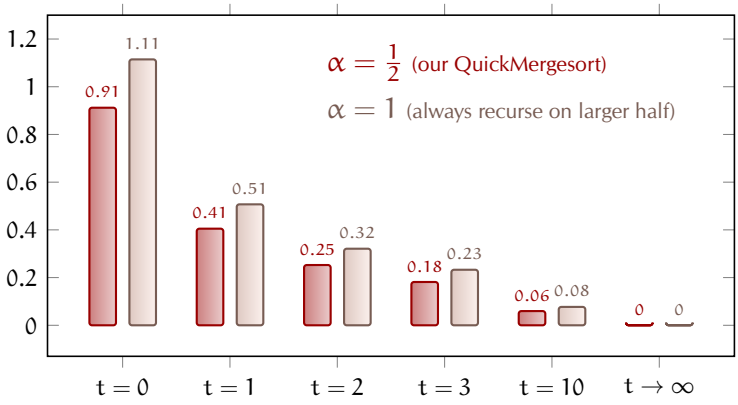
Result

Comparisons in QuickMergesort:

same as X ...

$$c(n) = x(n) + q_t \cdot n \pm O(\log n)$$

... + "QuickXsort penalty"



Conclusion

Findings

- QuickMergesort sorts **in-place** with close-to-**optimal #comparisons** (but not a stable sort)
- QuickXsort penalty rapidly decreases with sample size
- Skewed pivots **not** helpful (worse despite sorting a larger part by X!)

Extensions

- variance
- influence of fluctuations in Mergesort?
- limit distributions?

Conclusion

Findings

- QuickMergesort sorts **in-place** with close-to-**optimal #comparisons** (but not a stable sort)
- QuickXsort penalty rapidly decreases with sample size
- Skewed pivots **not** helpful (worse despite sorting a larger part by X!)

Extensions

- variance
- influence of fluctuations in Mergesort?
- limit distributions?

Conclusion

Findings

- QuickMergesort sorts **in-place** with close-to-**optimal #comparisons** (but not a stable sort)
- QuickXsort penalty rapidly decreases with sample size
- Skewed pivots **not** helpful (worse despite sorting a larger part by X!)

Extensions

- variance
- influence of fluctuations in Mergesort?
- limit distributions?

Conclusion

Findings

- QuickMergesort sorts **in-place** with close-to-**optimal #comparisons** (but not a stable sort)
- QuickXsort penalty rapidly decreases with sample size
- Skewed pivots **not** helpful (worse despite sorting a larger part by X!)

Extensions

- variance
- influence of fluctuations in Mergesort?
- limit distributions?

Conclusion

Findings

- QuickMergesort sorts **in-place** with close-to-**optimal #comparisons** (but not a stable sort)
- QuickXsort penalty rapidly decreases with sample size
- Skewed pivots **not** helpful (worse despite sorting a larger part by X!)

Extensions

- variance
- influence of fluctuations in Mergesort?
- limit distributions?

Conclusion

Findings

- QuickMergesort sorts **in-place** with close-to-**optimal #comparisons** (but not a stable sort)
- QuickXsort penalty rapidly decreases with sample size
- Skewed pivots **not** helpful (worse despite sorting a larger part by X!)

Extensions

- variance
- influence of fluctuations in Mergesort?
- limit distributions?

Conclusion

Findings

- QuickMergesort sorts **in-place** with close-to-**optimal #comparisons** (but not a stable sort)
- QuickXsort penalty rapidly decreases with sample size
- Skewed pivots **not** helpful (worse despite sorting a larger part by X!)

Extensions

- variance
- influence of fluctuations in Mergesort?
- limit distributions?


NO
MERGE
AREA

Icons made by *Freepik* and *Gregor Cresnar* from `www.flaticon.com`.
Title background by *Phil's 1stPix* (CC BY-BC-SA 2.0)