Sesquickselect

One and a Half Pivots for Cache-Efficient Selection

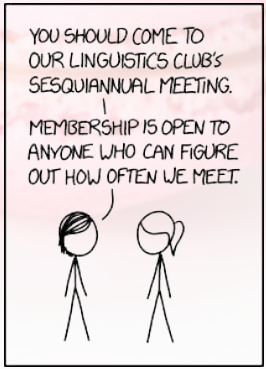
Sebastian Wild

wild@uwaterloo.ca

joint work with

Conrado Martínez and Markus Nebel

UNIVERSITY OF WATERLOO



ANALCO 2019

16th Meeting on Analytic Algorithmics and Combinatorics

Outline

1 Introduction

2 Random Ranks

3 Linear Ranks

4 Sesquickselect

5 Conclusion

What happened in quicksort?

- **since Java 7: dual-pivot** quicksort `Arrays.sort(int[])`



What happened in quicksort?

- **since Java 7: dual-pivot** quicksort `Arrays.sort(int[])`



- Why?

What happened in quicksort?

- **since Java 7: dual-pivot** quicksort `Arrays.sort(int[])`



- Why?  It is *faster!*

What happened in quicksort?

- **since Java 7: dual-pivot** quicksort `Arrays.sort(int[])`



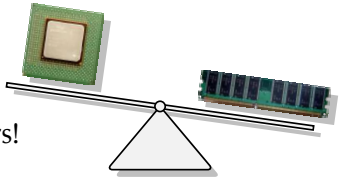
- Why?  It is *faster!*
- Why?

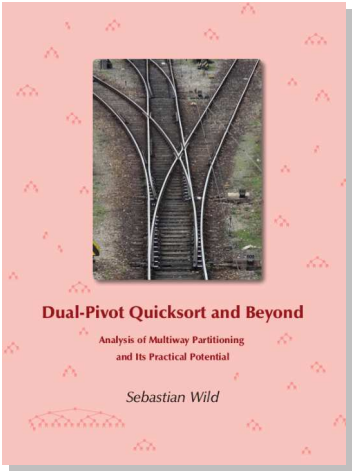
What happened in quicksort?

- **since Java 7: dual-pivot** quicksort `Arrays.sort(int[])`



- Why?  It is *faster!*
- Why? fewer memory transfers!



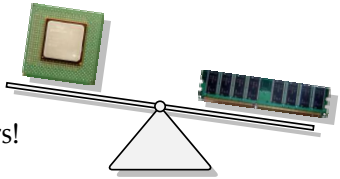


What happened in quicksort?

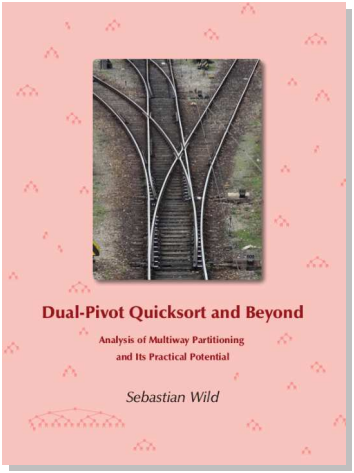
- **since Java 7: dual-pivot** quicksort `Arrays.sort(int[])`



- Why?  It is *faster!*
- Why? fewer memory transfers!



- **Intuition:** get more work done in one pass over data

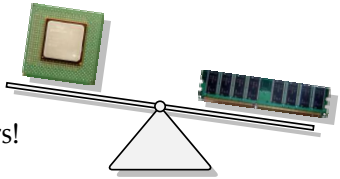


What happened in quicksort?

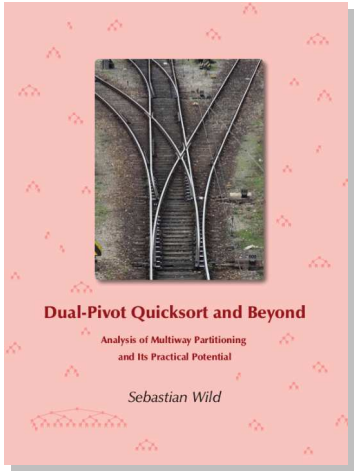
- **since Java 7:** **dual-pivot** quicksort `Arrays.sort(int[])`



- Why? It is *faster!*
- Why? fewer memory transfers!



- **Intuition:** get more work done in one pass over data





Is multiway partitioning also helpful for quickselect?

$\rightsquigarrow$ consider **memory transfer** volume! ← not comparisons!

Selection

The Selection Problem

Given an **unsorted** array `A[1..n]` of n elements and $m \in [1..n]$, find the **m-th smallest element** in `A`.

`A[m]` after sorting

Algorithms

- (obvious: `sort(A); return A[m];`)
- **Hoare 1961:** quickselect
- **Blum et al. 1973:** worst-case $O(n)$, large constant factor
- **Floyd & Rivest 1975:** asymptotically optimal #cmps (randomized in expectation)
  - quickselect with pivots from large sample $\Theta(n^{2/3}\log^{1/3}(n))$ elements

quickselect remained method of choice in programming libraries

Selection

The Selection Problem

Given an **unsorted** array `A[1..n]` of n elements and $m \in [1..n]$, find the **m-th smallest element** in `A`.

`A[m]` after sorting

Algorithms

- (**obvious:** `sort(A); return A[m];`)
- **Hoare 1961:** quickselect
- **Blum et al. 1973:** worst-case $O(n)$, large constant factor
- **Floyd & Rivest 1975:** asymptotically optimal #cmps (randomized in expectation)
  - quickselect with pivots from large sample $\Theta(n^{2/3}\log^{1/3}(n))$ elements

quickselect remained method of choice in programming libraries

Selection

The Selection Problem

Given an **unsorted** array `A[1..n]` of n elements and $m \in [1..n]$, find the **m-th smallest element** in `A`.

`A[m]` after sorting

Algorithms

- (**obvious:** `sort(A); return A[m];`)
- **Hoare 1961:** **quickselect**
- **Blum et al. 1973:** worst-case $O(n)$, large constant factor
- **Floyd & Rivest 1975:** asymptotically optimal #cmps (randomized in expectation)
  - quickselect with pivots from large sample $\Theta(n^{2/3}\log^{1/3}(n))$ elements

quickselect remained method of choice in programming libraries

Selection

The Selection Problem

Given an **unsorted** array `A[1..n]` of n elements and $m \in [1..n]$, find the **m-th smallest element** in `A`.

`A[m]` after sorting

Algorithms

- (**obvious:** `sort(A); return A[m];`)
- **Hoare 1961:** **quickselect**
- **Blum et al. 1973:** worst-case $O(n)$, large constant factor
- **Floyd & Rivest 1975:** asymptotically optimal #cmps (randomized in expectation)
  - quickselect with pivots from large sample $\Theta(n^{2/3}\log^{1/3}(n))$ elements

quickselect remained method of choice in programming libraries

Selection

The Selection Problem

Given an **unsorted** array `A[1..n]` of n elements and $m \in [1..n]$, find the **m-th smallest element** in `A`.

`A[m]` after sorting

Algorithms

- (**obvious:** `sort(A); return A[m];`)
- **Hoare 1961:** **quickselect**
- **Blum et al. 1973:** worst-case $O(n)$, large constant factor
- **Floyd & Rivest 1975:** asymptotically optimal #cmps (randomized in expectation)
  - quickselect with pivots from large sample $\Theta(n^{2/3}\log^{1/3}(n))$ elements

quickselect remained method of choice in programming libraries

Selection

The Selection Problem

Given an **unsorted** array `A[1..n]` of n elements and $m \in [1..n]$, find the **m-th smallest element** in `A`.

`A[m]` after sorting

Algorithms

- (**obvious:** `sort(A); return A[m];`)
- **Hoare 1961:** **quickselect**
- **Blum et al. 1973:** worst-case $O(n)$, large constant factor
- **Floyd & Rivest 1975:** asymptotically optimal #cmps (randomized in expectation)
  - quickselect with pivots from large sample $\Theta(n^{2/3}\log^{1/3}(n))$ elements

quickselect remained method of choice in programming libraries

Selection

The Selection Problem

Given an **unsorted** array `A[1..n]` of n elements and $m \in [1..n]$, find the **m-th smallest element** in `A`.

`A[m]` after sorting

Algorithms

- (**obvious:** `sort(A); return A[m];`)
- **Hoare 1961:** **quickselect**
- **Blum et al. 1973:** worst-case $O(n)$, large constant factor
- **Floyd & Rivest 1975:** asymptotically optimal #cmps (randomized in expectation)
  - quickselect with pivots from large sample $\Theta(n^{2/3}\log^{1/3}(n))$ elements



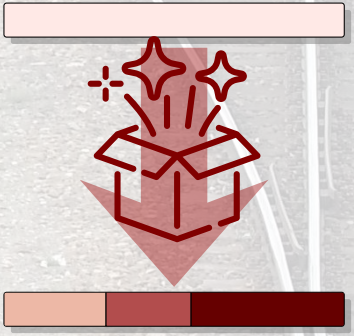
quickselect remained method of choice in programming libraries



YBB-partitioning



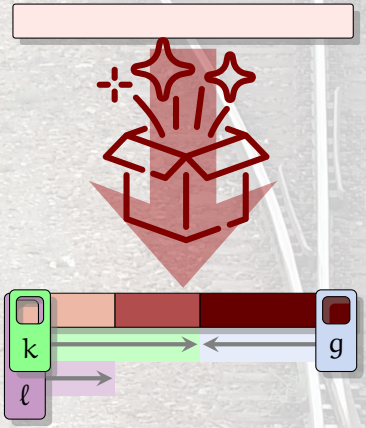
YBB-partitioning



Yaroslavskiy-Bentley-Bloch partitioning

- Split array into 3 regions: small, medium, large

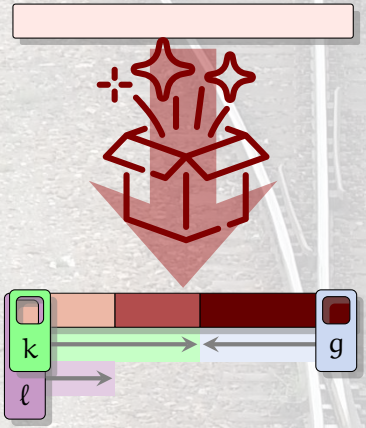
YBB-partitioning



Yaroslavskiy-Bentley-Bloch partitioning

- Split array into 3 regions: small , medium , large
- Uses three pointers k , g , ℓ

YBB-partitioning



Yaroslavskiy-Bentley-Bloch partitioning

- Split array into 3 regions: small, medium, large
- Uses three pointers k, g, l

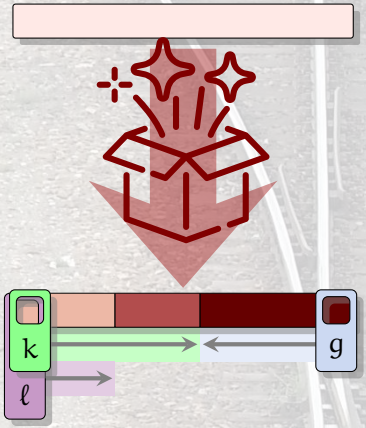
Costs:

$\propto$ cache misses

- count scanned *elements*

$$k + g + l$$

YBB-partitioning



Yaroslavskiy-Bentley-Bloch partitioning

- Split array into 3 regions: small, medium, large
- Uses three pointers k, g, ℓ

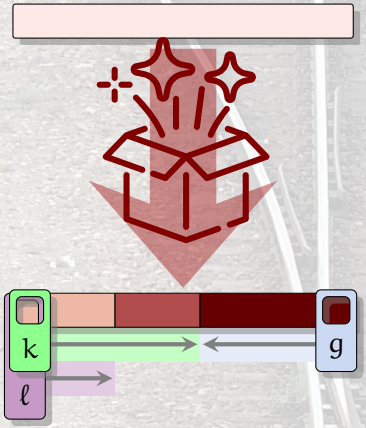
Costs:

$\propto$ cache misses

- count scanned *elements*

$$k + g + \ell = n + J_1$$

YBB-partitioning



Yaroslavskiy-Bentley-Bloch partitioning

- Split array into 3 regions: small, medium, large
- Uses three pointers k, g, ℓ

Costs:

$\propto$ cache misses

- count scanned *elements*

$$k + g + \ell = n + J_1$$

(comparisons more complicated $\rightsquigarrow$ ignore today)

Recurrence for costs

Dual-pivot quickselect

1. Choose pivots $P_1 < P_2$.
2. Partition around P_1, P_2 $\rightsquigarrow$
3. Recurse in one subproblem
   (or terminate if $m = R_{1,2}$).

Analysis

- $S_{n,m}$: **(random)** number of scanned elements to select mth smallest out of n
- **Recurrence** for the **distribution** of $S_{n,m}$:

$$S_{n,m} \overset{\mathcal{D}}{=} (n + J_1) + \begin{cases} S_{J_1, m} & \text{if } m \text{ lies in } \text{left} \text{ subarray} \\ S_{J_1, m-R_1} & \text{if } m \text{ lies in } \text{middle} \text{ subarray} \\ S_{J_1, m-R_2} & \text{if } m \text{ lies in } \text{right} \text{ subarray} \end{cases}$$

(partitioning costs)

Recurrence for costs

Dual-pivot quickselect

1. Choose pivots $P_1 < P_2$.
2. Partition around P_1, P_2 $\rightsquigarrow$ 

3. Recurse in one subproblem
(or terminate if $m = R_{1,2}$).

Analysis

- $S_{n,m}$: **(random)** number of scanned elements to select mth smallest out of n
- **Recurrence** for the **distribution** of $S_{n,m}$:

$$S_{n,m} \overset{D}{=} (n + J_1) + \begin{cases} S_{J_1,m} & \text{if } m \text{ lies in } \boxed{\text{left}} \text{ subarray} \\ S_{J_1,m-J_2} & \text{if } m \text{ lies in } \boxed{\text{middle}} \text{ subarray} \\ S_{J_1,m-R_{1,2}} & \text{if } m \text{ lies in } \boxed{\text{right}} \text{ subarray} \end{cases}$$

(partitioning costs)

Recurrence for costs

Dual-pivot quickselect

1. Choose pivots $P_1 < P_2$.
2. Partition around P_1, P_2 $\rightsquigarrow$ 

3. Recurse in one subproblem
   (or terminate if $m = R_{1,2}$).

Analysis

- $S_{n,m}$: **(random)** number of scanned elements to select mth smallest out of n
- **Recurrence** for the **distribution** of $S_{n,m}$:

$$S_{n,m} \overset{\mathcal{D}}{=} (n + J_1) + \begin{cases} S_{J_1, m} & \text{if } m \text{ lies in } \boxed{\text{left}} \text{ subarray} \\ S_{J_1, m-J_1-J_2} & \text{if } m \text{ lies in } \boxed{\text{middle}} \text{ subarray} \\ S_{J_1, n-m-J_2} & \text{if } m \text{ lies in } \boxed{\text{right}} \text{ subarray} \end{cases}$$

(partitioning costs)

Recurrence for costs

Dual-pivot quickselect

1. Choose pivots $P_1 < P_2$.
2. Partition around P_1, P_2 $\rightsquigarrow$
3. Recurse in one subproblem
   (or terminate if $m = R_{1,2}$).



Analysis

- $S_{n,m}$: (**random**) number of scanned elements to select mth smallest out of n
- **Recurrence** for the **distribution** of $S_{n,m}$:

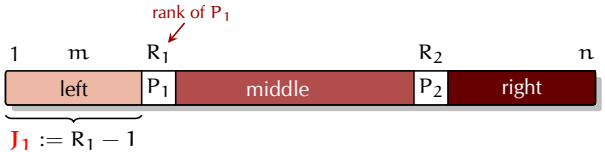
$$S_{n,m} \overset{\mathcal{D}}{=} (n + J_1) + \begin{cases} S_{J_1-1,m} & \text{if } m \text{ lies in } \text{left} \text{ subarray} \\ S_{J_2-J_1-1,m-J_1} & \text{if } m \text{ lies in } \text{middle} \text{ subarray} \\ S_{n-J_2,m-J_2} & \text{if } m \text{ lies in } \text{right} \text{ subarray} \end{cases}$$

(partitioning costs)

Recurrence for costs

Dual-pivot quickselect

1. Choose pivots $P_1 < P_2$.
2. Partition around P_1, P_2 $\rightsquigarrow$
3. Recurse in one subproblem
   (or terminate if $m = R_{1,2}$).



Analysis

- $S_{n,m}$: (**random**) number of scanned elements to select mth smallest out of n
- **Recurrence** for the **distribution** of $S_{n,m}$:

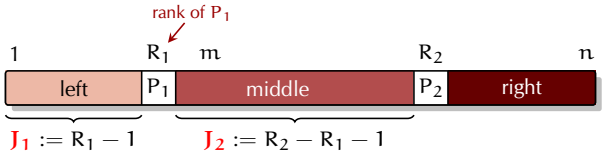
$$S_{n,m} \overset{\mathcal{D}}{=} (n + J_1) + \begin{cases} S_{J_1, m} & \text{if } m \text{ lies in } \text{left} \text{ subarray} \\ S_{J_2 - J_1 - 1, m - J_1 - 1} & \text{if } m \text{ lies in } \text{middle} \text{ subarray} \\ S_{n - J_2, m - J_2} & \text{if } m \text{ lies in } \text{right} \text{ subarray} \end{cases}$$

partitioning costs

Recurrence for costs

Dual-pivot quickselect

1. Choose pivots $P_1 < P_2$.
2. Partition around P_1, P_2 $\rightsquigarrow$
3. Recurse in one subproblem
   (or terminate if $m = R_{1,2}$).



Analysis

- $S_{n,m}$: (**random**) number of scanned elements to select mth smallest out of n
- **Recurrence** for the **distribution** of $S_{n,m}$:

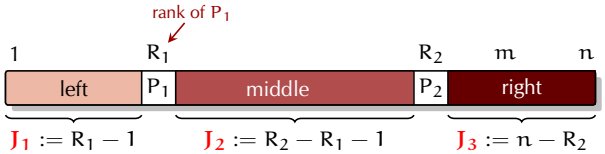
$$S_{n,m} \overset{\mathcal{D}}{=} (n + J_1) + \begin{cases} S_{J_1,m} & \text{if } m \text{ lies in } \boxed{\text{left}} \text{ subarray} \\ S_{J_2,m-R_1} & \text{if } m \text{ lies in } \boxed{\text{middle}} \text{ subarray} \\ S_{J_3,m-R_2} & \text{if } m \text{ lies in } \boxed{\text{right}} \text{ subarray} \end{cases}$$

(partitioning costs)

Recurrence for costs

Dual-pivot quickselect

1. Choose pivots $P_1 < P_2$.
2. Partition around P_1, P_2 $\rightsquigarrow$
3. Recurse in one subproblem
   (or terminate if $m = R_{1,2}$).



Analysis

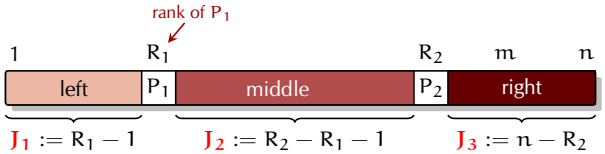
- $S_{n,m}$: (**random**) number of scanned elements to select mth smallest out of n
- **Recurrence** for the **distribution** of $S_{n,m}$:

$$S_{n,m} \overset{\mathcal{D}}{=} \underbrace{(n + J_1)}_{\text{partitioning costs}} + \begin{cases} S_{J_1,m} & \text{if } m \text{ lies in } \boxed{\text{left}} \text{ subarray} \\ S_{J_2,m-R_1} & \text{if } m \text{ lies in } \boxed{\text{middle}} \text{ subarray} \\ S_{J_3,m-R_2} & \text{if } m \text{ lies in } \boxed{\text{right}} \text{ subarray} \end{cases}$$

Recurrence for costs

Dual-pivot quickselect

1. Choose pivots $P_1 < P_2$.
2. Partition around P_1, P_2 $\rightsquigarrow$
3. Recurse in one subproblem
   (or terminate if $m = R_{1,2}$).



Analysis

- $S_{n,m}$: **(random)** number of scanned elements to select mth smallest out of n
- **Recurrence** for the **distribution** of $S_{n,m}$:

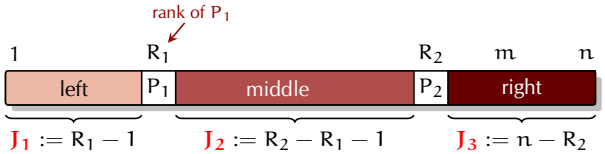
$$S_{n,m} \overset{\mathcal{D}}{=} (n + J_1) + \begin{cases} S_{J_1,m} & \text{if } m \text{ lies in left subarray} \\ S_{J_2,m-R_1} & \text{if } m \text{ lies in middle subarray} \\ S_{J_3,m-R_2} & \text{if } m \text{ lies in right subarray} \end{cases}$$

(partitioning costs)

Recurrence for costs

Dual-pivot quickselect

1. Choose pivots $P_1 < P_2$.
2. Partition around P_1, P_2 $\rightsquigarrow$
3. Recurse in one subproblem
   (or terminate if $m = R_{1,2}$).



Analysis

- $S_{n,m}$: **(random)** number of scanned elements to select mth smallest out of n
- **Recurrence** for the **distribution** of $S_{n,m}$:

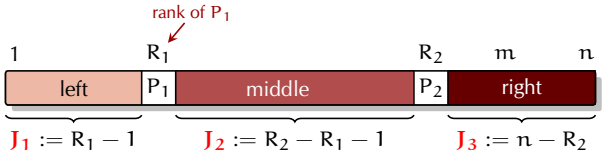
$$S_{n,m} \overset{\mathcal{D}}{=} (n + J_1) + \begin{cases} S_{J_1,m} & \text{if } m \text{ lies in left subarray} \\ S_{J_2,m-R_1} & \text{if } m \text{ lies in middle subarray} \\ S_{J_3,m-R_2} & \text{if } m \text{ lies in right subarray} \end{cases}$$

(partitioning costs)

Recurrence for costs

Dual-pivot quickselect

1. Choose pivots $P_1 < P_2$.
2. Partition around P_1, P_2 $\rightsquigarrow$
3. Recurse in one subproblem
   (or terminate if $m = R_{1,2}$).



Analysis

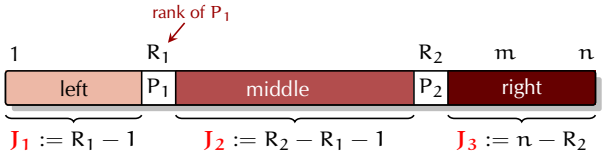
- $S_{n,m}$: **(random)** number of scanned elements to select mth smallest out of n
- **Recurrence** for the **distribution** of $S_{n,m}$:

$$S_{n,m} \overset{\mathcal{D}}{=} \underbrace{(n + J_1)}_{\text{partitioning costs}} + \begin{cases} S_{J_1,m} & \text{if } m \text{ lies in left subarray} \\ S_{J_2,m-R_1} & \text{if } m \text{ lies in middle subarray} \\ S_{J_3,m-R_2} & \text{if } m \text{ lies in right subarray} \end{cases}$$

Recurrence for costs

Dual-pivot quickselect

1. Choose pivots $P_1 < P_2$.
2. Partition around P_1, P_2 $\rightsquigarrow$
3. Recurse in one subproblem
   (or terminate if $m = R_{1,2}$).



Analysis

- $S_{n,m}$: **(random)** number of scanned elements to select mth smallest out of n
- **Recurrence** for the **distribution** of $S_{n,m}$:

$$S_{n,m} \overset{\mathcal{D}}{=} \underbrace{(n + J_1)}_{\text{partitioning costs}} + \begin{cases} S_{J_1,m} & \text{if } m \text{ lies in left subarray} \\ S_{J_2,m-R_1} & \text{if } m \text{ lies in middle subarray} \\ S_{J_3,m-R_2} & \text{if } m \text{ lies in right subarray} \end{cases}$$

$$\overset{\text{general case}}{=} T_{n,m} + \sum_{\ell=1}^{s} \mathbb{1}_{\{R_{\ell-1} < m < R_\ell\}} \cdot S_{J_\ell,\, m-R_{\ell-1}} \qquad (R_0 = 0,\ R_s = n+1)$$

(s: s segments)

Recurrence for costs

Dual-pivot quickselect

1. Choose pivots $P_1 < P_2$
2. Partition around P_1, P_2
3. Recurse in one su
   (or terminate if m

Analysis

- $S_{n,m}$: (ran
- **Recurrence**

$S_{n,m}$

general case

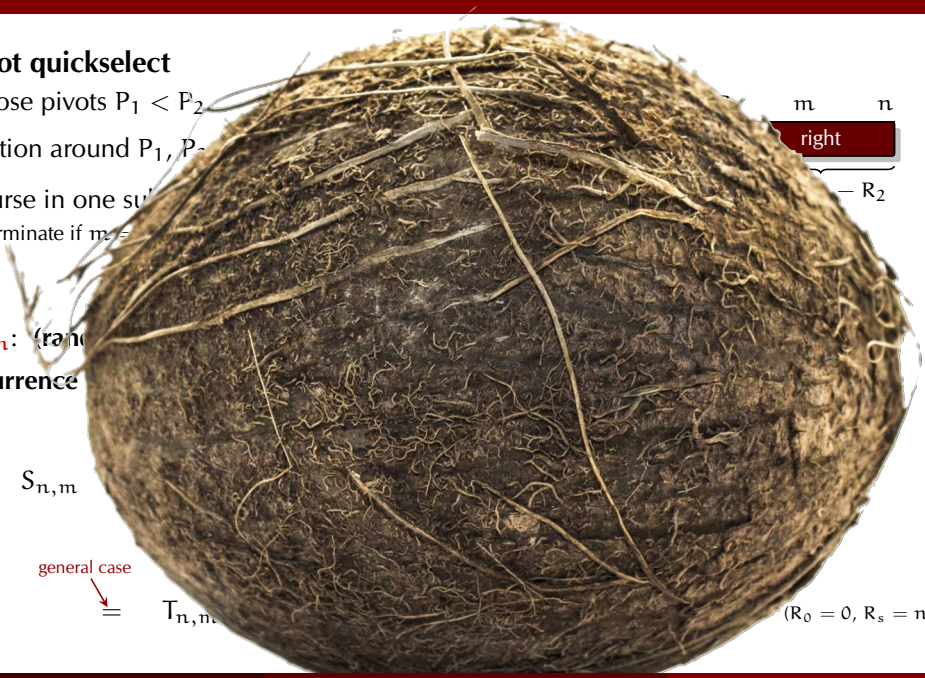
$= T_{n,m}$

m n

right

$- R_2$

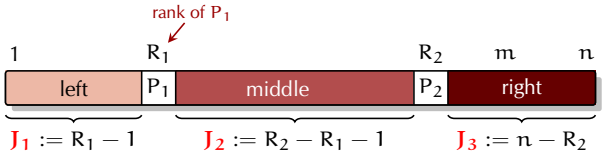
$(R_0 = 0, R_s = n + 1)$



Recurrence for costs

Dual-pivot quickselect

1. Choose pivots $P_1 < P_2$.
2. Partition around P_1, P_2 $\rightsquigarrow$
3. Recurse in one subproblem
   (or terminate if $m = R_{1,2}$).



Analysis

- $S_{n,m}$: **(random)** number of scanned elements to select mth smallest out of n
- **Recurrence** for the **distribution** of $S_{n,m}$:

$$S_{n,m} \overset{\mathcal{D}}{=} \underbrace{(n + J_1)}_{\text{partitioning costs}} + \begin{cases} S_{J_1,m} & \text{if } m \text{ lies in left subarray} \\ S_{J_2,m-R_1} & \text{if } m \text{ lies in middle subarray} \\ S_{J_3,m-R_2} & \text{if } m \text{ lies in right subarray} \end{cases}$$

$$\overset{\text{general case}}{=} T_{n,m} + \sum_{\ell=1}^{s} \mathbb{1}_{\{R_{\ell-1} < m < R_\ell\}} \cdot S_{J_\ell,\, m-R_{\ell-1}} \qquad (R_0 = 0,\ R_s = n+1)$$

(s: s segments)

Outline

1 Introduction

2 Random Ranks

3 Linear Ranks

4 Sesquickselect

5 Conclusion

Random-ranks recurrence



Complication:
2 parameters n, m

$\rightsquigarrow$

Simplification:
Random rank $m = M_n \overset{\mathcal{D}}{=} \text{Uniform}\{1, \dots, n\}$

- Write $\bar{S}_n := S_{n,M_n}$

$$\rightsquigarrow \bar{S}_n \overset{\mathcal{D}}{=} T_n + \sum_{\ell=1}^{s} \mathbb{1}_{\{R_{\ell-1} < m < R_\ell\}} \cdot S_{J_\ell,\ m-R_{\ell-1}}$$

Random-ranks recurrence



Complication:
2 parameters n, m

$\rightsquigarrow$

Simplification:
Random rank $m = M_n \overset{\mathcal{D}}{=} \text{Uniform}\{1, \dots, n\}$

- Write $\bar{S}_n := S_{n,M_n}$

$$\rightsquigarrow \bar{S}_n \overset{\mathcal{D}}{=} T_n + \sum_{\ell=1}^{s} \mathbb{1}_{\{R_{\ell-1} < m < R_\ell\}} \cdot S_{J_\ell,\ m-R_{\ell-1}}$$

Random-ranks recurrence



Complication:
2 parameters n, m

$\rightsquigarrow$

Simplification:
Random rank $m = M_n \overset{\mathcal{D}}{=} \text{Uniform}\{1, \dots, n\}$

- Write $\bar{S}_n := S_{n,M_n}$

$$\rightsquigarrow \bar{S}_n \overset{\mathcal{D}}{=} T_n + \sum_{\ell=1}^{s} \mathbb{1}_{\{R_{\ell-1} < m < R_\ell\}} \cdot S_{J_\ell,\ m-R_{\ell-1}}$$

Random-ranks recurrence



Complication:
2 parameters n, m

$\rightsquigarrow$

Simplification:
Random rank $m = M_n \overset{\mathcal{D}}{=} \text{Uniform}\{1, \dots, n\}$

- Write $\bar{S}_n := S_{n,M_n}$

$$\rightsquigarrow \bar{S}_n \overset{\mathcal{D}}{=} T_n + \sum_{\ell=1}^{s} \mathbb{1}_{\{R_{\ell-1} < M_n < R_\ell\}} \cdot S_{J_\ell,\ m - R_{\ell-1}}$$

Random-ranks recurrence



Complication:
2 parameters n, m

$\rightsquigarrow$

Simplification:
Random rank $m = M_n \overset{\mathcal{D}}{=} \text{Uniform}\{1, \dots, n\}$

- Write $\bar{S}_n := S_{n, M_n}$

$$\rightsquigarrow \bar{S}_n \overset{\mathcal{D}}{=} T_n + \sum_{\ell=1}^{s} \mathbb{1}_{\{R_{\ell-1} < M_n < R_\ell\}} \cdot S_{J_\ell,\, M_n - R_{\ell-1}}$$

Random-ranks recurrence



Complication:
2 parameters n, m

$\rightsquigarrow$

Simplification:
Random rank $m = M_n \overset{\mathcal{D}}{=} \text{Uniform}\{1, \dots, n\}$

- Write $\bar{S}_n := S_{n,M_n}$

$$\rightsquigarrow \bar{S}_n \overset{\mathcal{D}}{=} T_n + \sum_{\ell=1}^{s} \mathbb{1}_{\{R_{\ell-1} < M_n < R_\ell\}} \cdot S_{J_\ell, M_n - R_{\ell-1}}$$

conditionally uniform

Random-ranks recurrence



Complication:
2 parameters n, m

$\rightsquigarrow$

Simplification:
Random rank $m = M_n \overset{\mathcal{D}}{=} \text{Uniform}\{1, \dots, n\}$

- Write $\bar{S}_n := S_{n,M_n}$

$$\rightsquigarrow \bar{S}_n \overset{\mathcal{D}}{=} T_n + \sum_{\ell=1}^{s} \mathbb{1}_{\{R_{\ell-1} < M_n < R_\ell\}} \cdot \bar{S}_{J_\ell}$$

Random-ranks recurrence



Complication:
2 parameters n, m

$\rightsquigarrow$

Simplification:
Random rank $m = M_n \overset{\mathcal{D}}{=} \text{Uniform}\{1, \dots, n\}$

- Write $\bar{S}_n := S_{n,M_n}$

$\rightsquigarrow$ $\bar{S}_n \overset{\mathcal{D}}{=} T_n + \sum_{\ell=1}^{s} \mathbb{1}_{\{R_{\ell-1} < M_n < R_\ell\}} \cdot \bar{S}_{J_\ell}$ $\rightsquigarrow$ single-parameter recurrence!

Random-ranks recurrence



Complication:
2 parameters n, m

$\rightsquigarrow$

Simplification:
Random rank $m = M_n \overset{\mathcal{D}}{=} \text{Uniform}\{1, \dots, n\}$

- Write $\bar{S}_n := S_{n,M_n}$

$\rightsquigarrow$ $\bar{S}_n \overset{\mathcal{D}}{=} T_n + \sum_{\ell=1}^{s} \mathbb{1}_{\{R_{\ell-1} < M_n < R_\ell\}} \cdot \bar{S}_{J_\ell}$ $\rightsquigarrow$ single-parameter recurrence!

- **same recurrence for**
  - $s \geqslant 2$ of segments
  - pivots chosen from sample (constant sample size k, fixed order statistics in sample $\vec{t}$)

Random-ranks recurrence



Complication:
2 parameters n, m

$\rightsquigarrow$

Simplification:
Random rank $m = M_n \overset{\mathcal{D}}{=} \text{Uniform}\{1, \dots, n\}$

- Write $\bar{S}_n := S_{n,M_n}$

$\rightsquigarrow \bar{S}_n \overset{\mathcal{D}}{=} T_n + \sum_{\ell=1}^{s} \mathbb{1}_{\{R_{\ell-1} < M_n < R_\ell\}} \cdot \bar{S}_{J_\ell}$ $\rightsquigarrow$ single-parameter recurrence!

- **same recurrence for**
  - $s \geqslant 2$ of segments
  - pivots chosen from sample (constant sample size k, fixed order statistics in sample $\vec{t}$)

Random-ranks recurrence



Complication:
2 parameters n, m

$\rightsquigarrow$

Simplification:
Random rank $m = M_n \overset{\mathcal{D}}{=} \text{Uniform}\{1, \dots, n\}$

- Write $\bar{S}_n := S_{n,M_n}$

$\rightsquigarrow \bar{S}_n \overset{\mathcal{D}}{=} T_n + \sum_{\ell=1}^{s} \mathbb{1}_{\{R_{\ell-1} < M_n < R_\ell\}} \cdot \bar{S}_{J_\ell}$ $\rightsquigarrow$ single-parameter recurrence!

- **same recurrence for**
  - $s \geqslant 2$ of segments
  - pivots chosen from sample (constant sample size k, fixed order statistics in sample $\vec{t}$)

Random-ranks recurrence



Complication:
2 parameters n, m

$\rightsquigarrow$

Simplification:
Random rank $m = M_n \overset{\mathcal{D}}{=} \text{Uniform}\{1, \dots, n\}$

- Write $\bar{S}_n := S_{n,M_n}$

$\rightsquigarrow \bar{S}_n \overset{\mathcal{D}}{=} T_n + \sum_{\ell=1}^{s} \mathbb{1}_{\{R_{\ell-1} < M_n < R_\ell\}} \cdot \bar{S}_{J_\ell}$ $\rightsquigarrow$ single-parameter recurrence!

- **same recurrence for**
  - $s \geqslant 2$ of segments
  - pivots chosen from sample (constant sample size k, fixed order statistics in sample $\vec{t}$)

    influences dist. of R_ℓ, J_ℓ

Random-ranks recurrence



Complication:
2 parameters n, m

$\rightsquigarrow$

Simplification:
Random rank $m = M_n \overset{\mathcal{D}}{=} \text{Uniform}\{1, \dots, n\}$

- Write $\bar{S}_n := S_{n,M_n}$

$\rightsquigarrow$ $\bar{S}_n \overset{\mathcal{D}}{=} T_n + \sum_{\ell=1}^{s} \mathbb{1}_{\{R_{\ell-1} < M_n < R_\ell\}} \cdot \bar{S}_{J_\ell}$ $\rightsquigarrow$ single-parameter recurrence!

- **same recurrence for**
  - $s \geqslant 2$ of segments
  - pivots chosen from sample (constant sample size k, fixed order statistics in sample $\vec{t}$)

    influences dist. of R_ℓ, J_ℓ

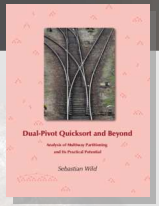
$\rightsquigarrow$ asymptotic for $\mathbb{E}[\bar{S}_n]$ using *distributional master theorem*

Distributional Master Theorem



Distributional Master Theorem

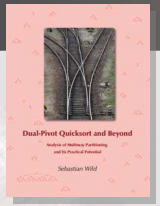
Distributional Master Theorem (DMT) (Theorem 2.76)



Yields asymptotic approximation for $\mathbb{E}[C_n]$ when

Distributional Master Theorem

Distributional Master Theorem (DMT) (Theorem 2.76)



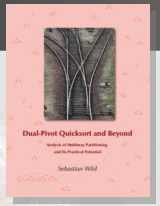
Yields asymptotic approximation for $\mathbb{E}[C_n]$ when

1. C_n satisfies D & C-recurrence with random subproblem sizes

$$C_n \overset{\mathcal{D}}{=} T_n + \sum_{r=1}^{s} A_r^{(n)} \cdot C_{J_r^{(n)}}^{(r)}$$

Distributional Master Theorem

Distributional Master Theorem (DMT) (Theorem 2.76)



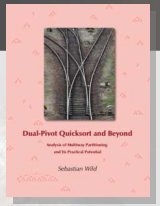
Yields asymptotic approximation for $\mathbb{E}[C_n]$ when

1. C_n satisfies D & C-recurrence with random subproblem sizes
   $$C_n \overset{\mathcal{D}}{=} T_n + \sum_{r=1}^{s} A_r^{(n)} \cdot C_{J_r^{(n)}}^{(r)}$$
2. $\mathbb{E}[T_n]$ from "standard scale"
   $$\mathbb{E}[T_n] \sim K n^{\alpha} \ln^{\beta}(n)$$

Distributional Master Theorem

Distributional Master Theorem (DMT)

(Theorem 2.76)



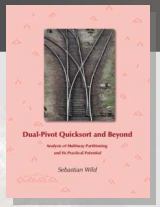
Yields asymptotic approximation for $\mathbb{E}[C_n]$ when

1. C_n satisfies D & C-recurrence with random subproblem sizes
   $$C_n \overset{\mathcal{D}}{=} T_n + \sum_{r=1}^{s} A_r^{(n)} \cdot C_{J_r^{(n)}}^{(r)}$$
2. $\mathbb{E}[T_n]$ from "standard scale"
   $$\mathbb{E}[T_n] \sim K n^{\alpha} \ln^{\beta}(n)$$
3. relative subproblem sizes fulfill local limit law with smooth density
   $$n \cdot \mathbb{P}\left[\frac{J_r^{(n)}}{n} \in \left(z - \tfrac{1}{n}, z\right]\right] \approx f_r(z)$$

Distributional Master Theorem

Distributional Master Theorem (DMT)

(Theorem 2.76)



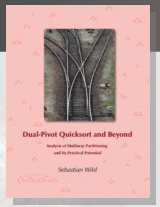
Yields asymptotic approximation for $\mathbb{E}[C_n]$ when

1. C_n satisfies D & C-recurrence with random subproblem sizes
   $$C_n \overset{\mathcal{D}}{=} T_n + \sum_{r=1}^{s} A_r^{(n)} \cdot C_{J_r^{(n)}}^{(r)}$$
2. $\mathbb{E}[T_n]$ from "standard scale"
   $$\mathbb{E}[T_n] \sim K n^{\alpha} \ln^{\beta}(n)$$
3. relative subproblem sizes fulfill local limit law with smooth density
   $$n \cdot \mathbb{P}\left[\frac{J_r^{(n)}}{n} \in \left(z - \tfrac{1}{n}, z\right]\right] \approx f_r(z)$$
4. coefficients A_r are smooth in z conditional on $Z_r = z$
   $$\mathbb{E}\left[A_r^{(n)} \,\middle|\, \frac{J_r^{(n)}}{n} \in \left(z - \tfrac{1}{n}, z\right]\right] \approx a_r(z)$$

Distributional Master Theorem

Distributional Master Theorem (DMT)

(Theorem 2.76)



Yields asymptotic approximation for $\mathbb{E}[C_n]$ when

1. C_n satisfies D & C-recurrence with random subproblem sizes
$$C_n \overset{\mathcal{D}}{=} T_n + \sum_{r=1}^{s} A_r^{(n)} \cdot C_{J_r^{(n)}}^{(r)}$$
2. $\mathbb{E}[T_n]$ from "standard scale"
$$\mathbb{E}[T_n] \sim K n^{\alpha} \ln^{\beta}(n)$$
3. relative subproblem sizes fulfill local limit law with smooth density
$$n \cdot \mathbb{P}\left[\frac{J_r^{(n)}}{n} \in \left(z - \tfrac{1}{n}, z\right]\right] \approx f_r(z)$$

Hölder continuous

4. coefficients A_r are smooth in z conditional on $Z_r = z$
$$\mathbb{E}\left[A_r^{(n)} \,\middle|\, \frac{J_r^{(n)}}{n} \in \left(z - \tfrac{1}{n}, z\right]\right] \approx a_r(z)$$

Results for random ranks

Results

1. YBB quickselect **11% less** scanned elements than standard quickselect (no pivot sampling)
2. more than 2 pivots **not** helpful (Care needed for fair comparison → paper.)

⇝ full results for random ranks available
including pivot sampling, e.g., median of 3

Results for random ranks

Results

1. YBB quickselect **11% less** scanned elements than standard quickselect (no pivot sampling)
2. more than 2 pivots **not** helpful (Care needed for fair comparison → paper.)

⇝ full results for random ranks available
including pivot sampling, e.g., median of 3

Results for random ranks

Results

25% for sorting!

1. YBB quickselect **11% less** scanned elements than standard quickselect (no pivot sampling)
2. more than 2 pivots **not** helpful (Care needed for fair comparison → paper.)

⇝ full results for random ranks available
including pivot sampling, e.g., median of 3

Results for random ranks

Results

25% for sorting!

1. YBB quickselect **11% less** scanned elements than standard quickselect (no pivot sampling)
2. more than 2 pivots **not** helpful (Care needed for fair comparison $\rightarrow$ paper.)

$\rightsquigarrow$ full results for random ranks available
including pivot sampling, e.g., median of 3

Results for random ranks

Results

25% for sorting!

1. YBB quickselect **11% less** scanned elements than standard quickselect (no pivot sampling)
2. more than 2 pivots **not** helpful (Care needed for fair comparison $\rightarrow$ paper.)

unlike sorting

$\rightsquigarrow$ full results for random ranks available
including pivot sampling, e.g., median of 3

Results for random ranks

Results

25% for sorting!

1. YBB quickselect **11% less** scanned elements than standard quickselect (no pivot sampling)
2. more than 2 pivots **not** helpful (Care needed for fair comparison $\rightarrow$ paper.)

unlike sorting

$\rightsquigarrow$ full results for random ranks available
including pivot sampling, e.g., median of 3

Results for random ranks

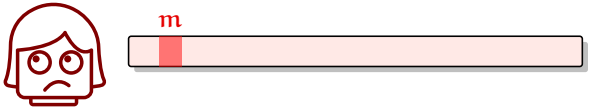
Results

25% for sorting!

1. YBB quickselect **11% less** scanned elements than standard quickselect (no pivot sampling)
2. more than 2 pivots **not** helpful (Care needed for fair comparison → paper.)

unlike sorting

⇝ full results for random ranks available
including pivot sampling, e.g., median of 3



⇝ **Adapt** pivot choice to sought rank
(as in Floyd-Rivest, but small sample)

Results for random ranks

Results

25% for sorting!

1. YBB quickselect **11% less** scanned elements than standard quickselect (no pivot sampling)
2. more than 2 pivots **not** helpful (Care needed for fair comparison $\rightarrow$ paper.)

unlike sorting

$\rightsquigarrow$ full results for random ranks available
including pivot sampling, e.g., median of 3

why shoot for median?



$\rightsquigarrow$ **Adapt** pivot choice to sought rank
(as in Floyd-Rivest, but small sample)

Results for random ranks

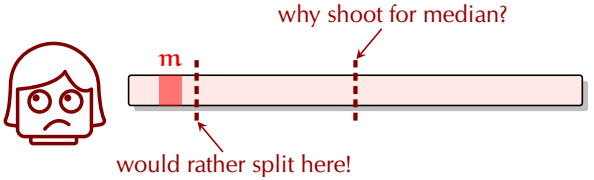
Results

25% for sorting!

1. YBB quickselect **11% less** scanned elements than standard quickselect (no pivot sampling)
2. more than 2 pivots **not** helpful (Care needed for fair comparison → paper.)

unlike sorting

⇝ full results for random ranks available
including pivot sampling, e.g., median of 3



⇝ **Adapt** pivot choice to sought rank
(as in Floyd-Rivest, but small sample)

Results for random ranks

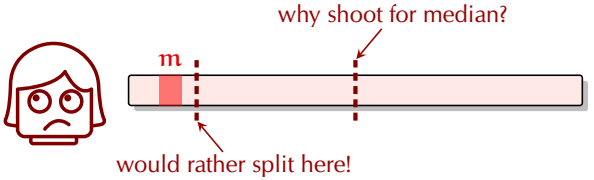
Results

25% for sorting!

1. YBB quickselect **11% less** scanned elements than standard quickselect (no pivot sampling)
2. more than 2 pivots **not** helpful (Care needed for fair comparison → paper.)

unlike sorting

⇝ full results for random ranks available
including pivot sampling, e.g., median of 3



⇝ **Adapt** pivot choice to sought rank
(as in Floyd-Rivest, but small sample)

Results for random ranks

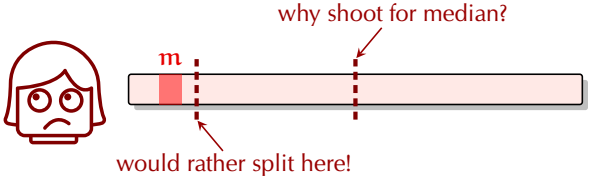
Results

25% for sorting!

1. YBB quickselect **11% less** scanned elements than standard quickselect (no pivot sampling)
2. more than 2 pivots **not** helpful (Care needed for fair comparison → paper.)

unlike sorting

⇝ full results for random ranks available
including pivot sampling, e.g., median of 3



⇝ **Adapt** pivot choice to sought rank
(as in Floyd-Rivest, but small sample)

Random-ranks analysis only works for **fixed pivot sampling choice**

Results for random ranks

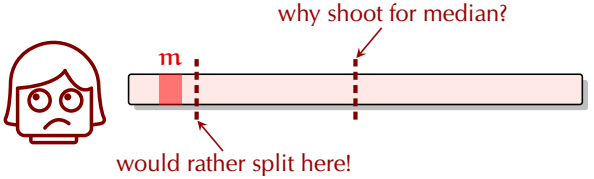
Results

25% for sorting!

1. YBB quickselect **11% less** scanned elements than standard quickselect (no pivot sampling)
2. more than 2 pivots **not** helpful (Care needed for fair comparison → paper.)

unlike sorting

⇝ full results for random ranks available
including pivot sampling, e.g., median of 3



⇝ ~~**Adapt** pivot choice to sought rank~~
~~(as in Floyd-Rivest, but small sample)~~

Random-ranks analysis only works for **fixed pivot sampling choice**

inevitably learn more about m
than in which subproblem it lies

Outline

1 Introduction

2 Random Ranks

3 Linear Ranks

4 Sesquickselect

5 Conclusion

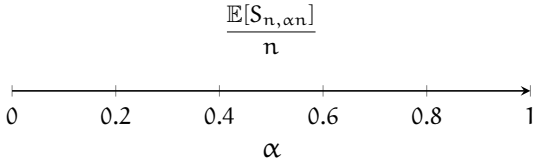
Linear ranks



back to 2 parameters …

consider $\boxed{m = \alpha \cdot n}$ and $n \to \infty$

relative rank $\in (0, 1)$



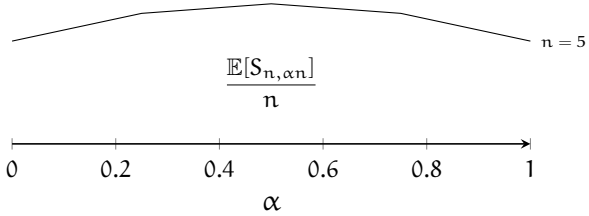
Linear ranks



back to 2 parameters ...

consider $m = \alpha \cdot n$ and $n \to \infty$

relative rank $\in (0, 1)$



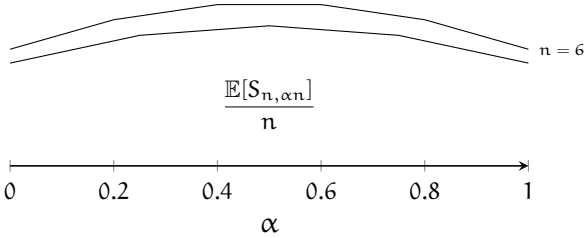
Linear ranks



back to 2 parameters …

consider $\boxed{m = \alpha \cdot n}$ and $n \to \infty$

relative rank $\in (0, 1)$



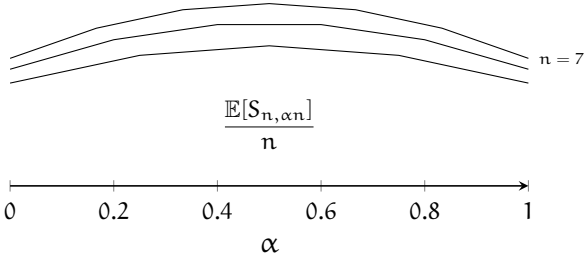
Linear ranks



back to 2 parameters ...

consider $\boxed{m = \alpha \cdot n}$ and $n \to \infty$

relative rank $\in (0, 1)$



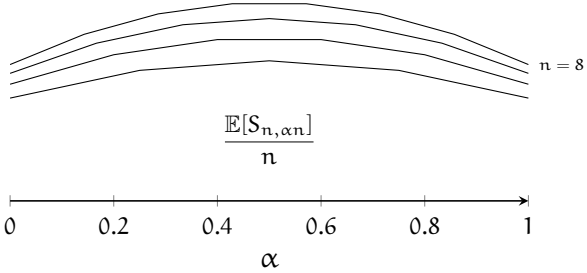
Linear ranks



back to 2 parameters …

consider $\boxed{m = \alpha \cdot n}$ and $n \to \infty$

relative rank $\in (0, 1)$



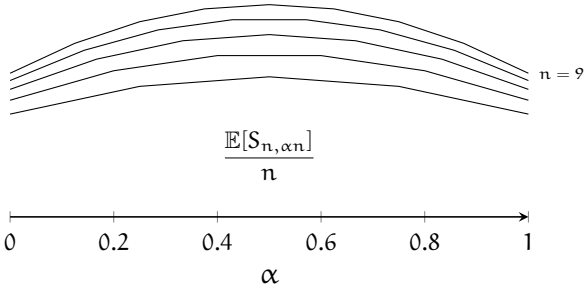
Linear ranks



back to 2 parameters …

consider $m = \alpha \cdot n$ and $n \to \infty$

relative rank $\in (0, 1)$



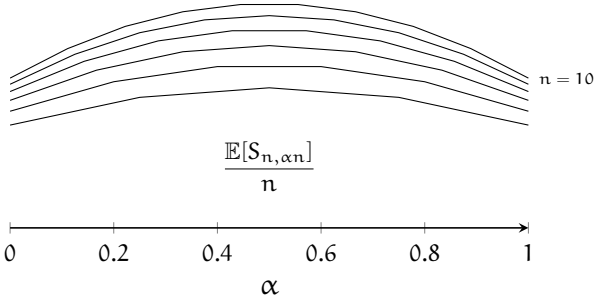
Linear ranks



back to 2 parameters …

consider $\boxed{m = \alpha \cdot n}$ and $n \to \infty$

relative rank $\in (0, 1)$



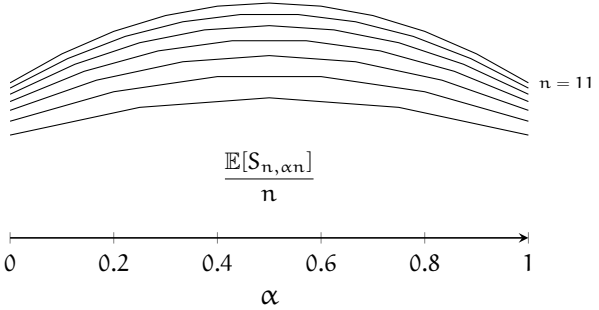
Linear ranks



back to 2 parameters ...

consider $\boxed{m = \alpha \cdot n}$ and $n \to \infty$

relative rank $\in (0, 1)$



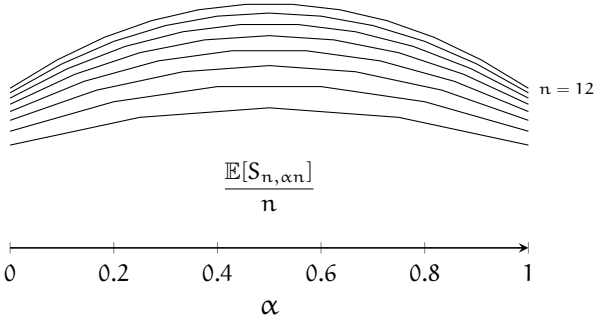
Linear ranks



back to 2 parameters …

consider $\boxed{m = \alpha \cdot n}$ and $n \to \infty$

relative rank $\in (0, 1)$



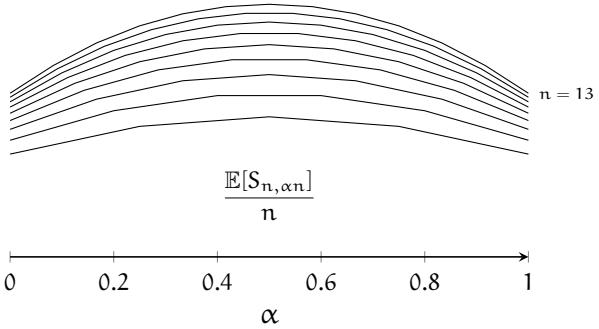
Linear ranks



back to 2 parameters …

consider $\boxed{m = \alpha \cdot n}$ and $n \to \infty$

relative rank $\in (0, 1)$



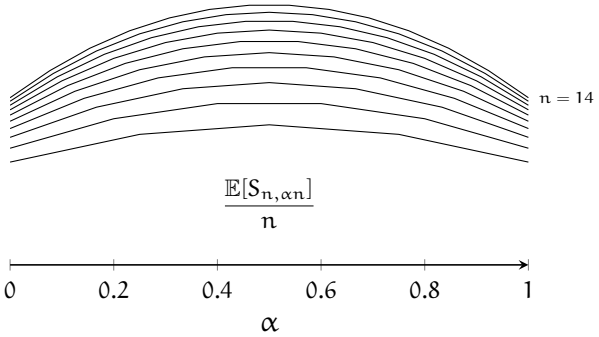
Linear ranks



back to 2 parameters …

consider $\boxed{m = \alpha \cdot n}$ and $n \to \infty$

relative rank $\in (0, 1)$



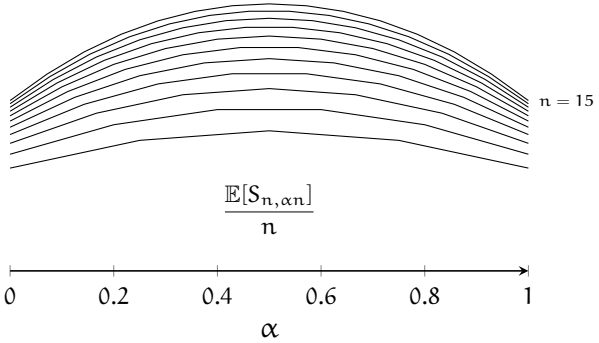
Linear ranks



back to 2 parameters ...

consider $\boxed{m = \alpha \cdot n}$ and $n \to \infty$

relative rank $\in (0, 1)$



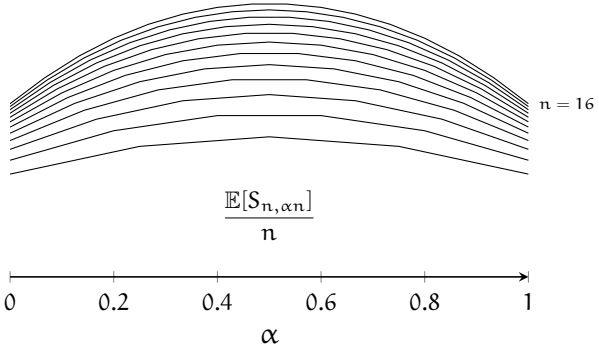
Linear ranks



back to 2 parameters ...

consider $\boxed{m = \alpha \cdot n}$ and $n \to \infty$

relative rank $\in (0, 1)$



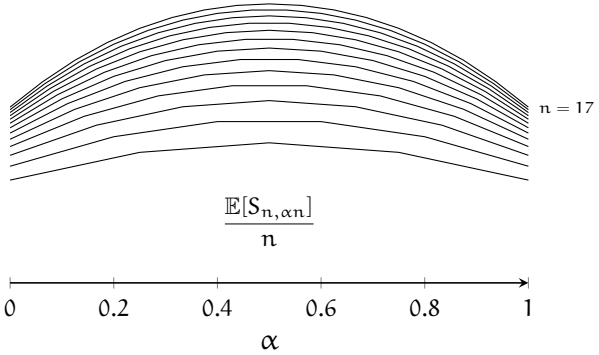
Linear ranks



back to 2 parameters …

consider $\boxed{m = \alpha \cdot n}$ and $n \to \infty$

relative rank $\in (0, 1)$



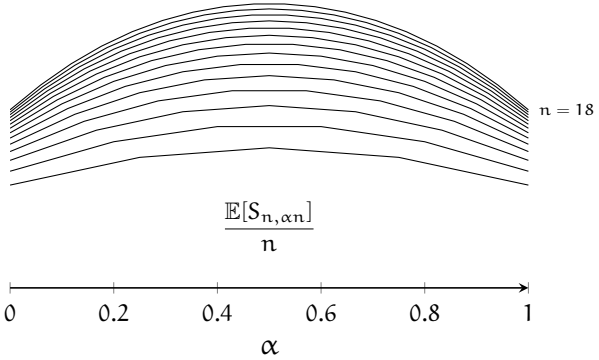
Linear ranks



back to 2 parameters …

consider $\boxed{m = \alpha \cdot n}$ and $n \to \infty$

relative rank $\in (0, 1)$

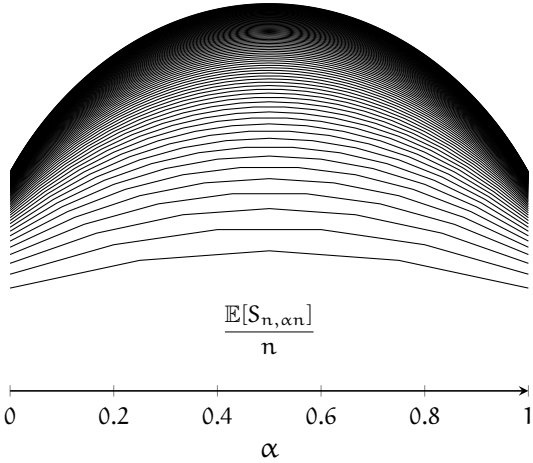


Linear ranks



back to 2 parameters ... consider $\boxed{m = \alpha \cdot n}$ and $n \to \infty$

relative rank $\in (0, 1)$



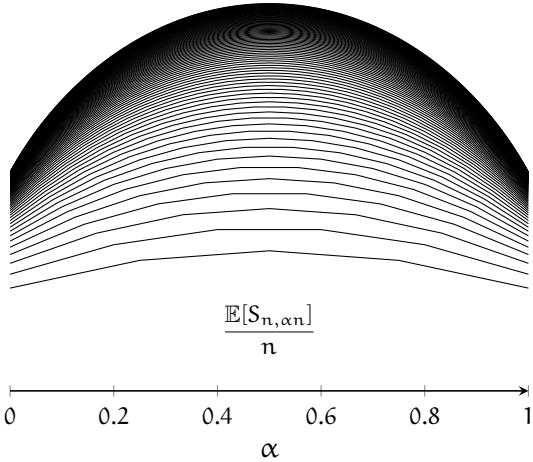
Linear ranks



back to 2 parameters ...

consider $m = \alpha \cdot n$ and $n \to \infty$

relative rank $\in (0, 1)$



Can we find $s(\alpha) = \lim_{n\to\infty} \frac{\mathbb{E}[S_{n,\alpha n}]}{n}$?

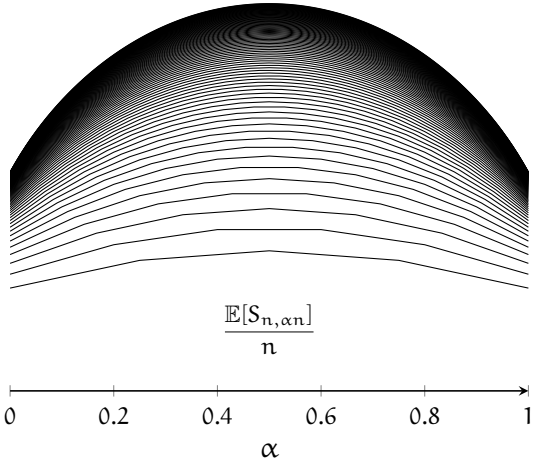
Linear ranks



back to 2 parameters …

consider $\boxed{m = \alpha \cdot n}$ and $n \to \infty$

relative rank $\in (0, 1)$



Can we find $s(\alpha) = \lim_{n\to\infty} \frac{\mathbb{E}[S_{n,\alpha n}]}{n}$?

Two steps:

1. Prove convergence
2. Compute limiting function

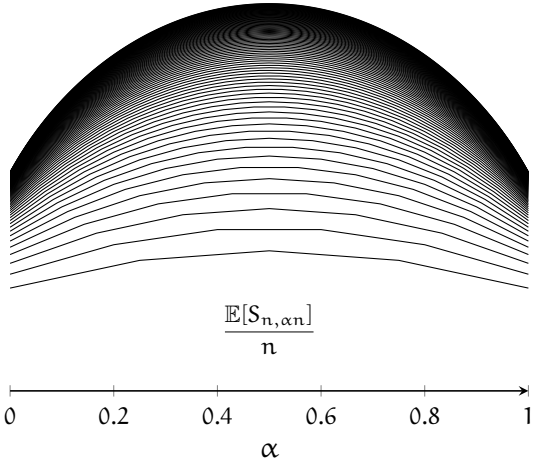
Linear ranks



back to 2 parameters ...

consider $\boxed{m = \alpha \cdot n}$ and $n \to \infty$

relative rank $\in (0, 1)$



Can we find $s(\alpha) = \lim_{n\to\infty} \frac{\mathbb{E}[S_{n,\alpha n}]}{n}$?

Two steps:

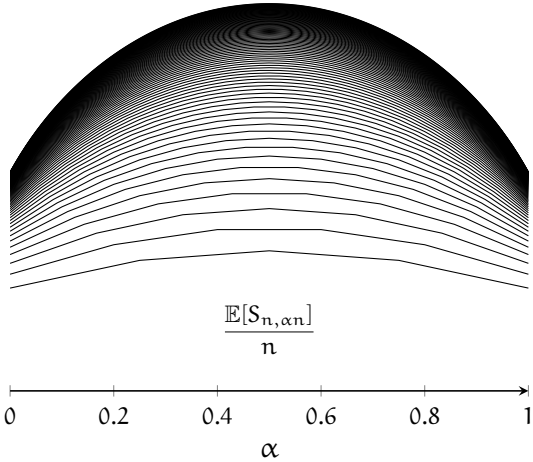
1. Prove convergence
2. Compute limiting function

Linear ranks



back to 2 parameters ... consider $\boxed{m = \alpha \cdot n}$ and $n \to \infty$

relative rank $\in (0, 1)$



Can we find $s(\alpha) = \lim_{n\to\infty} \frac{\mathbb{E}[S_{n,\alpha n}]}{n}$?

Two steps:

1. Prove convergence
2. Compute limiting function

An $\int$-equation for $s(\alpha)$

Rewriting the recurrence:

$$\frac{\mathbb{E}[S_{n,m}]}{n} = \frac{\mathbb{E}[T_{n,m}]}{n} + \sum_{\ell=1}^{s} \sum_{\underline{r}=1}^{m-1} \sum_{\overline{r}=m+1}^{n} \frac{\overline{r}-\underline{r}-1}{n} \cdot \mathbb{P}\big[(R_{\ell-1}, R_\ell) = (\underline{r}, \overline{r})\big] \cdot \frac{\mathbb{E}[S_{\overline{r}-\underline{r}-1, m-\underline{r}}]}{\overline{r}-\underline{r}-1}$$

$\rightsquigarrow$ *Assuming convergence, $s(\alpha)$ given by this integral-equation!*

An $\int$-equation for $s(\alpha)$

Rewriting the recurrence:

norm. cost

$$\frac{\mathbb{E}[S_{n,m}]}{n} = \frac{\mathbb{E}[T_{n,m}]}{n} + \sum_{\ell=1}^{s} \sum_{\underline{r}=1}^{m-1} \sum_{\overline{r}=m+1}^{n} \frac{\overline{r}-\underline{r}-1}{n} \cdot \mathbb{P}\big[(R_{\ell-1}, R_\ell) = (\underline{r}, \overline{r})\big] \cdot \frac{\mathbb{E}[S_{\overline{r}-\underline{r}-1,\, m-\underline{r}}]}{\overline{r}-\underline{r}-1}$$

$\rightsquigarrow$ *Assuming convergence, $s(\alpha)$ given by this integral-equation!*

An $\int$-equation for $s(\alpha)$

Rewriting the recurrence:

norm. cost — norm. partitioning cost

$$\frac{\mathbb{E}[S_{n,m}]}{n} = \frac{\mathbb{E}[T_{n,m}]}{n} + \sum_{\ell=1}^{s} \sum_{\underline{r}=1}^{m-1} \sum_{\overline{r}=m+1}^{n} \frac{\overline{r}-\underline{r}-1}{n} \cdot \mathbb{P}\big[(R_{\ell-1}, R_\ell) = (\underline{r}, \overline{r})\big] \cdot \frac{\mathbb{E}[S_{\overline{r}-\underline{r}-1, m-\underline{r}}]}{\overline{r}-\underline{r}-1}$$

$\rightsquigarrow$ *Assuming convergence, $s(\alpha)$ given by this integral-equation!*

An $\int$-equation for $s(\alpha)$

Rewriting the recurrence:

norm. cost — norm. partitioning cost — all ranges $\underline{r}, \overline{r}$

$$\frac{\mathbb{E}[S_{n,m}]}{n} = \frac{\mathbb{E}[T_{n,m}]}{n} + \sum_{\ell=1}^{s} \sum_{\underline{r}=1}^{m-1} \sum_{\overline{r}=m+1}^{n} \frac{\overline{r}-\underline{r}-1}{n} \cdot \mathbb{P}\big[(R_{\ell-1}, R_\ell) = (\underline{r}, \overline{r})\big] \cdot \frac{\mathbb{E}[S_{\overline{r}-\underline{r}-1, m-\underline{r}}]}{\overline{r}-\underline{r}-1}$$

$\rightsquigarrow$ *Assuming convergence, $s(\alpha)$ given by this integral-equation!*

An $\int$-equation for $s(\alpha)$

Rewriting the recurrence:

norm. cost

norm. partitioning cost

all ranges $\underline{r}, \overline{r}$

prob for subproblem $(\underline{r}, \overline{r})$

$$\frac{\mathbb{E}[S_{n,m}]}{n} = \frac{\mathbb{E}[T_{n,m}]}{n} + \sum_{\ell=1}^{s} \sum_{\underline{r}=1}^{m-1} \sum_{\overline{r}=m+1}^{n} \frac{\overline{r}-\underline{r}-1}{n} \cdot \mathbb{P}\big[(R_{\ell-1}, R_\ell) = (\underline{r}, \overline{r})\big] \cdot \frac{\mathbb{E}[S_{\overline{r}-\underline{r}-1, m-\underline{r}}]}{\overline{r}-\underline{r}-1}$$

$\rightsquigarrow$ *Assuming convergence, $s(\alpha)$ given by this integral-equation!*

An $\int$-equation for $s(\alpha)$

Rewriting the recurrence:

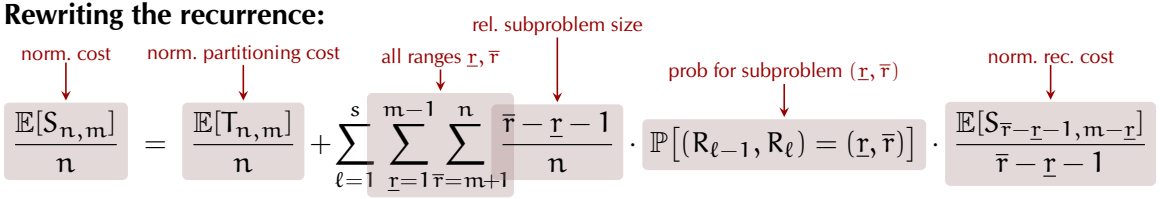
norm. cost — norm. partitioning cost — all ranges $\underline{r}, \overline{r}$ — prob for subproblem $(\underline{r}, \overline{r})$ — norm. rec. cost

$$\frac{\mathbb{E}[S_{n,m}]}{n} = \frac{\mathbb{E}[T_{n,m}]}{n} + \sum_{\ell=1}^{s} \sum_{\underline{r}=1}^{m-1} \sum_{\overline{r}=m+1}^{n} \frac{\overline{r}-\underline{r}-1}{n} \cdot \mathbb{P}\big[(R_{\ell-1}, R_\ell) = (\underline{r}, \overline{r})\big] \cdot \frac{\mathbb{E}[S_{\overline{r}-\underline{r}-1, m-\underline{r}}]}{\overline{r}-\underline{r}-1}$$

$\rightsquigarrow$ *Assuming convergence, $s(\alpha)$ given by this integral-equation!*

An $\int$-equation for $s(\alpha)$

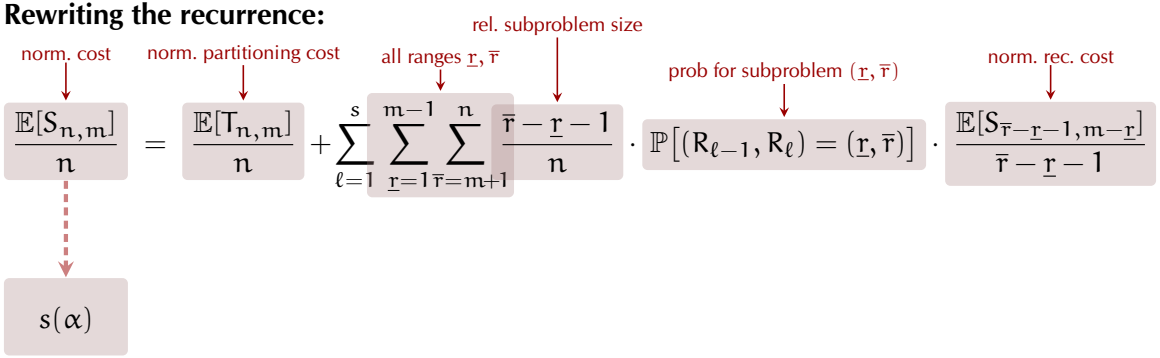
Rewriting the recurrence:



$\rightsquigarrow$ *Assuming convergence, $s(\alpha)$ given by this integral-equation!*

An $\int$-equation for $s(\alpha)$

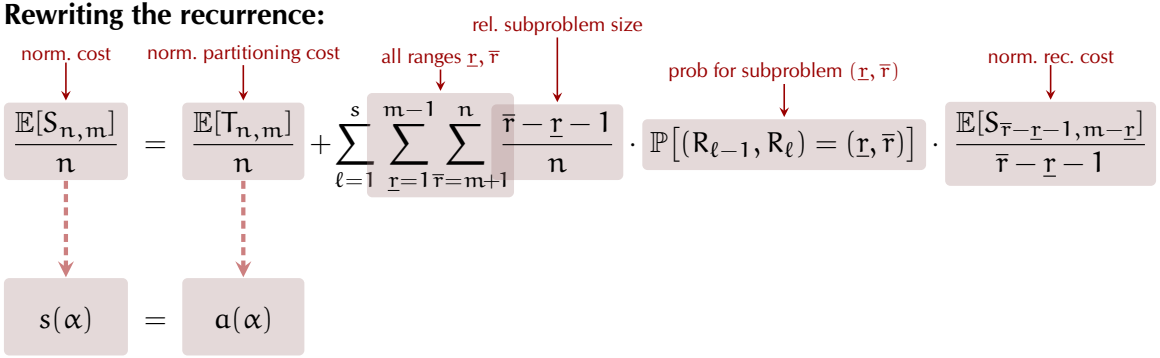
Rewriting the recurrence:



$\rightsquigarrow$ *Assuming convergence, $s(\alpha)$ given by this integral-equation!*

An $\int$-equation for $s(\alpha)$

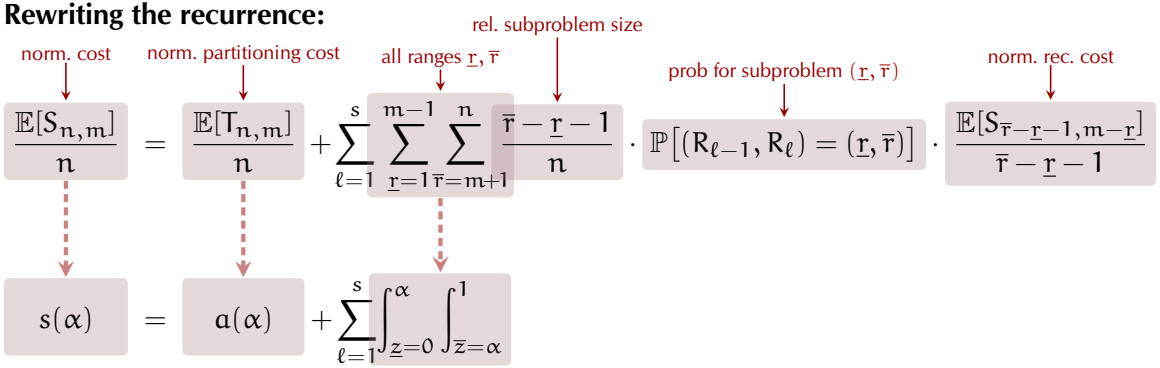
Rewriting the recurrence:



$\rightsquigarrow$ *Assuming convergence, $s(\alpha)$ given by this integral-equation!*

An $\int$-equation for $s(\alpha)$

Rewriting the recurrence:



$\rightsquigarrow$ *Assuming convergence, $s(\alpha)$ given by this integral-equation!*

An $\int$-equation for $s(\alpha)$

Rewriting the recurrence:

norm. cost — norm. partitioning cost — all ranges $\underline{r}, \overline{r}$ — rel. subproblem size — prob for subproblem $(\underline{r}, \overline{r})$ — norm. rec. cost

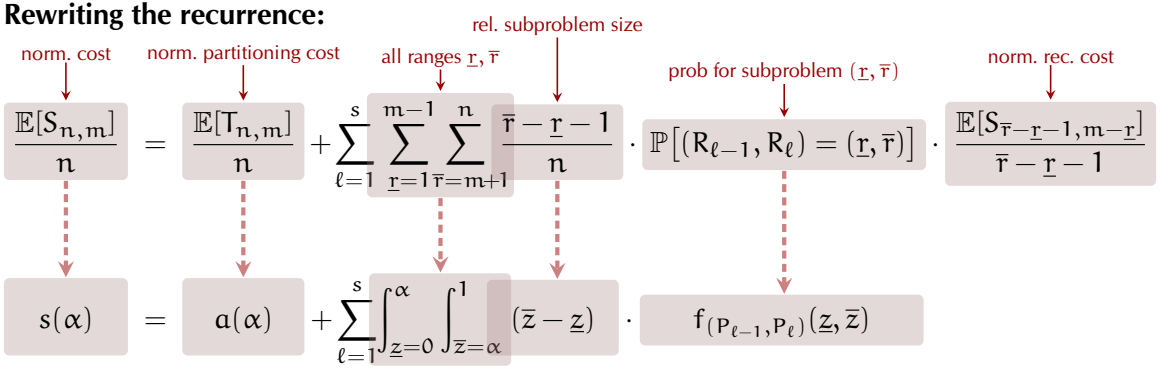
$$\frac{\mathbb{E}[S_{n,m}]}{n} = \frac{\mathbb{E}[T_{n,m}]}{n} + \sum_{\ell=1}^{s} \sum_{\underline{r}=1}^{m-1} \sum_{\overline{r}=m+1}^{n} \frac{\overline{r}-\underline{r}-1}{n} \cdot \mathbb{P}\big[(R_{\ell-1}, R_\ell) = (\underline{r}, \overline{r})\big] \cdot \frac{\mathbb{E}[S_{\overline{r}-\underline{r}-1,\, m-\underline{r}}]}{\overline{r}-\underline{r}-1}$$

$$s(\alpha) = a(\alpha) + \sum_{\ell=1}^{s} \int_{\underline{z}=0}^{\alpha} \int_{\overline{z}=\alpha}^{1} (\overline{z}-\underline{z})$$

⇝ *Assuming convergence, $s(\alpha)$ given by this integral-equation!*

An $\int$-equation for $s(\alpha)$

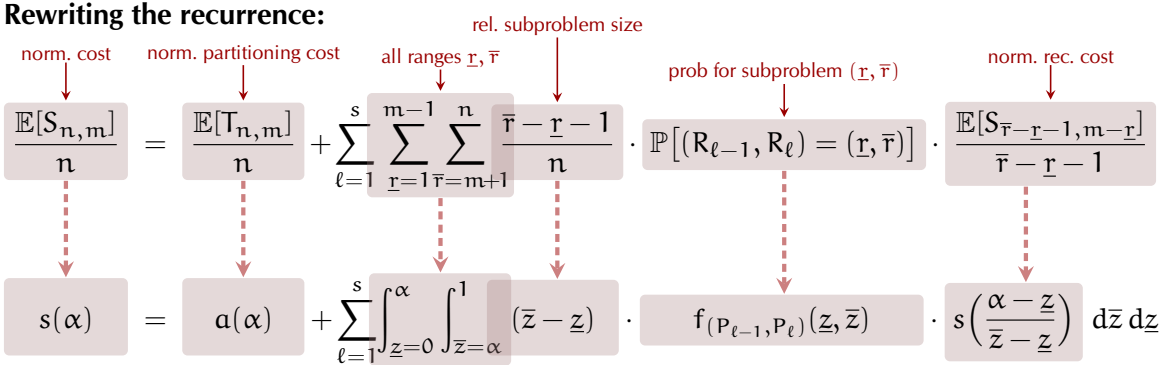
Rewriting the recurrence:



⇝ *Assuming convergence, $s(\alpha)$ given by this integral-equation!*

An $\int$-equation for $s(\alpha)$

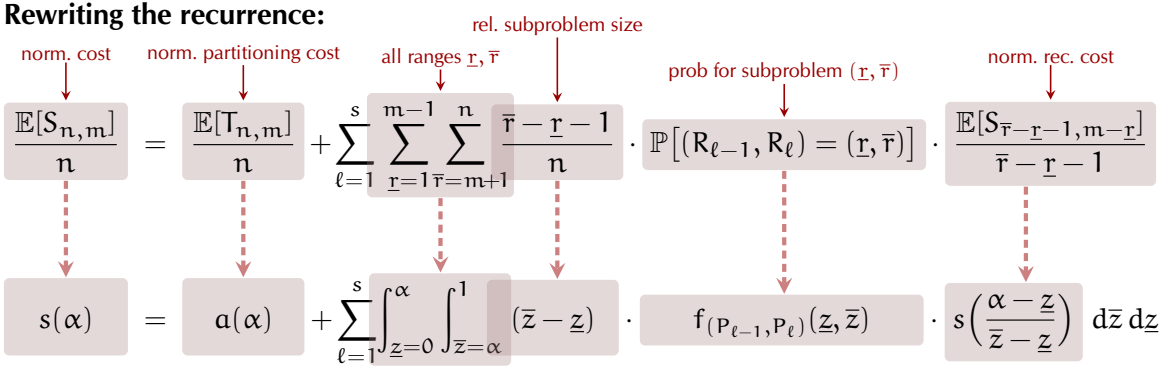
Rewriting the recurrence:



⇝ *Assuming convergence, $s(\alpha)$ given by this integral-equation!*

An $\int$-equation for $s(\alpha)$

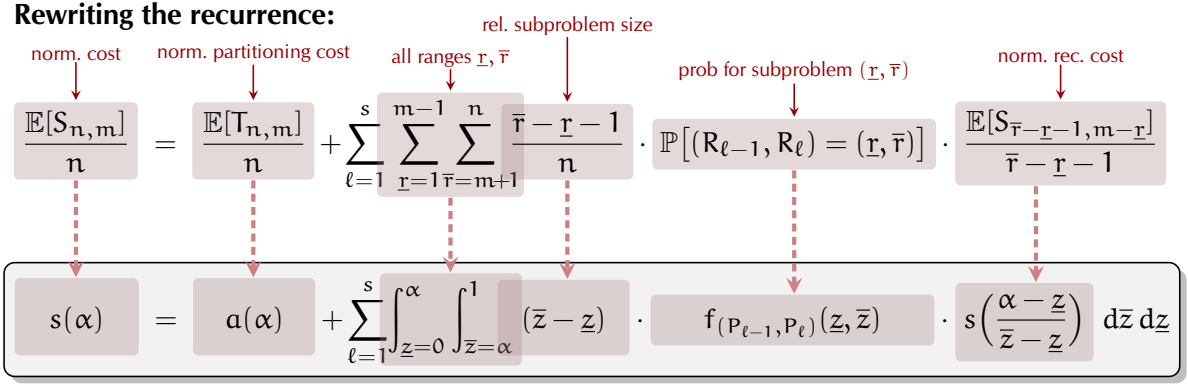
Rewriting the recurrence:



$\rightsquigarrow$ *Assuming convergence, $s(\alpha)$ given by this integral-equation!*

An $\int$-equation for $s(\alpha)$

Rewriting the recurrence:



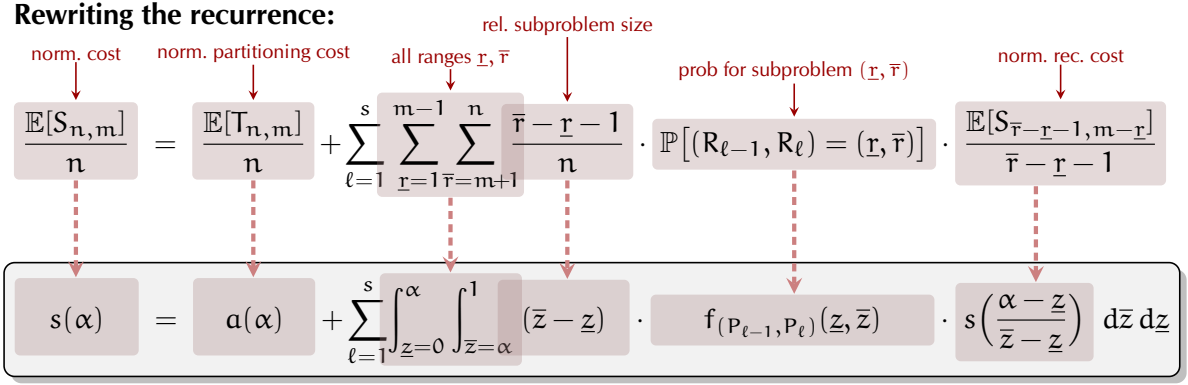
⇝ *Assuming convergence, $s(\alpha)$ given by this integral-equation!*

Proving convergence:

- **previously:** (only) with heavy probabilistic machinery (showing stronger statements)
- **here:** elementary real-analysis proof doable!
- works for all parameters choices (s, k, pivot sampling)

An $\int$-equation for $s(\alpha)$

Rewriting the recurrence:



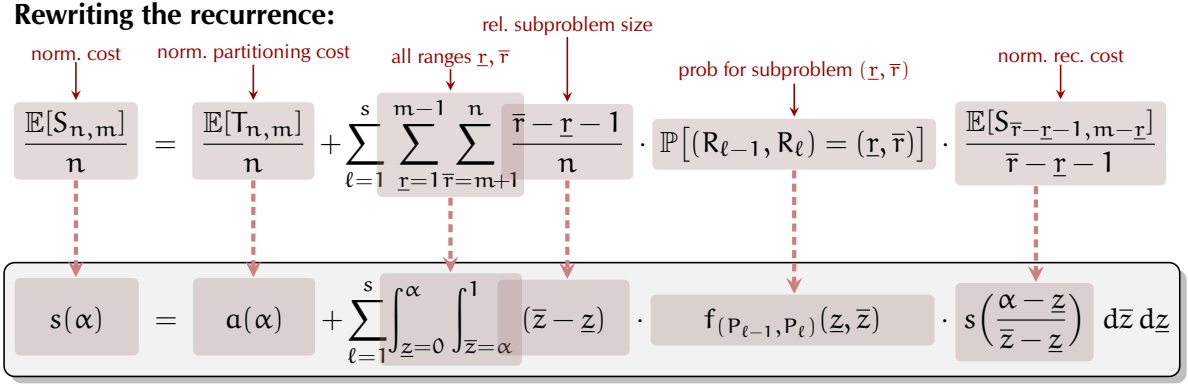
$\rightsquigarrow$ *Assuming convergence, $s(\alpha)$ given by this integral-equation!*

Proving convergence:

- **previously:** (only) with heavy probabilistic machinery (showing stronger statements)
- **here:** elementary real-analysis proof doable!
- works for all parameters choices (s, k, pivot sampling)

An $\int$-equation for $s(\alpha)$

Rewriting the recurrence:



⇝ *Assuming convergence, $s(\alpha)$ given by this integral-equation!*

Proving convergence:

- **previously:** (only) with heavy probabilistic machinery (showing stronger statements)
- **here:** elementary real-analysis proof doable!

works for all parameters choices (s, k, pivot sampling)

An $\int$-equation for $s(\alpha)$

Rewriting the recurrence:

norm. cost — norm. partitioning cost — all ranges $\underline{r}, \overline{r}$ — rel. subproblem size — prob for subproblem $(\underline{r}, \overline{r})$ — norm. rec. cost

$$\frac{\mathbb{E}[S_{n,m}]}{n} = \frac{\mathbb{E}[T_{n,m}]}{n} + \sum_{\ell=1}^{s} \sum_{\underline{r}=1}^{m-1} \sum_{\overline{r}=m+1}^{n} \frac{\overline{r}-\underline{r}-1}{n} \cdot \mathbb{P}\big[(R_{\ell-1}, R_\ell) = (\underline{r}, \overline{r})\big] \cdot \frac{\mathbb{E}[S_{\overline{r}-\underline{r}-1,\, m-\underline{r}}]}{\overline{r}-\underline{r}-1}$$

$$s(\alpha) = a(\alpha) + \sum_{\ell=1}^{s} \int_{\underline{z}=0}^{\alpha} \int_{\overline{z}=\alpha}^{1} (\overline{z}-\underline{z}) \cdot f_{(P_{\ell-1}, P_\ell)}(\underline{z}, \overline{z}) \cdot s\left(\frac{\alpha-\underline{z}}{\overline{z}-\underline{z}}\right) \mathrm{d}\overline{z}\, \mathrm{d}\underline{z}$$

$\rightsquigarrow$ *Assuming convergence, $s(\alpha)$ given by this integral-equation!*

Proving convergence:

- **previously:** (only) with heavy probabilistic machinery (showing stronger statements)
- **here:** elementary real-analysis proof doable!



works for all parameters choices (s, k, pivot sampling)

Convergence

1 Prove convergence

Goal: Bound $r_{n,m} := \left|\mathbb{E}[S_{n,m}] - s(\frac{m}{n})n\right|$ (we show: $O(n^{1-\varepsilon})$)

- find recurrence for $r_{n,m}$
  - same shape as for $\mathbb{E}[S_{n,m}]$
  - smaller toll function $b_{r,m}$ $\rightarrow$ essentially $\left|\sum\sum - \int\int\right|$.
- still two parameters, but now a crude bound suffices

Convergence

1 Prove convergence

Goal: Bound $r_{n,m} := \left|\mathbb{E}[S_{n,m}] - s(\frac{m}{n})n\right|$ (we show: $O(n^{1-\varepsilon})$)

- find recurrence for $r_{n,m}$
  - same shape as for $\mathbb{E}[S_{n,m}]$
  - smaller toll function $b_{r,m}$ $\rightarrow$ essentially $\left|\sum\sum - \int\int\right|$.
- still two parameters, but now a crude bound suffices

Convergence

1 Prove convergence

Goal: Bound $r_{n,m} := \left|\mathbb{E}[S_{n,m}] - s(\frac{m}{n})n\right|$ (we show: $O(n^{1-\varepsilon})$)

- find recurrence for $r_{n,m}$
  - same shape as for $\mathbb{E}[S_{n,m}]$
  - smaller toll function $b_{r,m}$ $\rightarrow$ essentially $\left|\sum\sum - \int\int\right|$.
- still two parameters, but now a crude bound suffices

Convergence

1 Prove convergence

Goal: Bound $r_{n,m} := \left|\mathbb{E}[S_{n,m}] - s(\frac{m}{n})n\right|$ (we show: $O(n^{1-\varepsilon})$)

- find recurrence for $r_{n,m}$
  - same shape as for $\mathbb{E}[S_{n,m}]$
  - smaller toll function $b_{r,m}$ $\rightarrow$ essentially $\left|\sum\sum - \int\int\right|$.
- still two parameters, but now a crude bound suffices

The gory details



1. bounding difference of sum and integral
   - split range of integration
     - where transform varies to quickly
     - where we hit discontinuities
   - use local limit law for Dirichlet-multinomial distribution
2. bound $r_{n,m}$
   - bound expected subproblem size (geometrically decreasing)

   $\rightsquigarrow$ dominated by toll function

Solving the integral equation

2 Compute limiting function

$$s(\alpha) \;=\; a(\alpha) \;+\; \sum_{\ell=1}^{s} \int_{\underline{z}=0}^{\alpha} \int_{\overline{z}=\alpha}^{1} (\overline{z}-\underline{z}) \cdot f_{(P_{\ell-1},P_\ell)}(\underline{z},\overline{z}) \cdot s\Big(\frac{\alpha-\underline{z}}{\overline{z}-\underline{z}}\Big)\, d\overline{z}\, d\underline{z}$$

- massage integrals with substitutions and take derivatives

$\rightsquigarrow$ higher-order differential equation

Solving the integral equation

2 Compute limiting function

$$s(\alpha) \;=\; a(\alpha) \;+\; \sum_{\ell=1}^{s} \int_{\underline{z}=0}^{\alpha} \int_{\overline{z}=\alpha}^{1} (\overline{z}-\underline{z}) \cdot f_{(P_{\ell-1},P_\ell)}(\underline{z},\overline{z}) \cdot s\Big(\frac{\alpha-\underline{z}}{\overline{z}-\underline{z}}\Big)\, d\overline{z}\, d\underline{z}$$

- massage integrals with substitutions and take derivatives

$\rightsquigarrow$ higher-order differential equation

Solving the integral equation

2 Compute limiting function

$$s(\alpha) \;=\; a(\alpha) \;+\; \sum_{\ell=1}^{s} \int_{\underline{z}=0}^{\alpha} \int_{\overline{z}=\alpha}^{1} (\overline{z}-\underline{z}) \cdot f_{(P_{\ell-1},P_\ell)}(\underline{z},\overline{z}) \cdot s\Big(\frac{\alpha-\underline{z}}{\overline{z}-\underline{z}}\Big)\, d\overline{z}\, d\underline{z}$$

- massage integrals with substitutions and take derivatives

$\rightsquigarrow$ higher-order differential equation

Solving the integral equation

2 Compute limiting function

$$s(\alpha) \;=\; a(\alpha) \;+\; \sum_{\ell=1}^{s} \int_{\underline{z}=0}^{\alpha} \int_{\overline{z}=\alpha}^{1} (\overline{z}-\underline{z}) \cdot f_{(P_{\ell-1},P_\ell)}(\underline{z},\overline{z}) \cdot s\Big(\frac{\alpha-\underline{z}}{\overline{z}-\underline{z}}\Big)\, d\overline{z}\, d\underline{z}$$

- massage integrals with substitutions and take derivatives

$\rightsquigarrow$ higher-order differential equation



general case quite intractable, but

- analytic solutions for simplest cases
- good numeric approximation

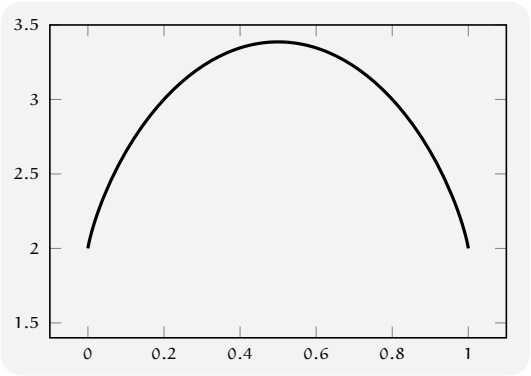
Linear-ranks Results



Analytic solutions for simplest cases

- **standard quickselect** $s(\alpha) = 2\,h(\alpha) + 2$

$$h(\alpha) = -\alpha \ln(\alpha) - (1-\alpha)\ln(1-\alpha)$$



Linear-ranks Results



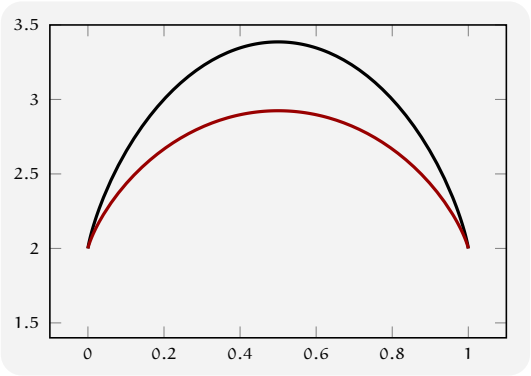
Analytic solutions for simplest cases

- **standard quickselect** $s(\alpha) = 2\,h(\alpha) + 2$

$$h(\alpha) = -\alpha \ln(\alpha) - (1-\alpha)\ln(1-\alpha)$$



YBB-quickselect $s(\alpha) = 1.\overline{3}\,h(\alpha) + 2$



Linear-ranks Results



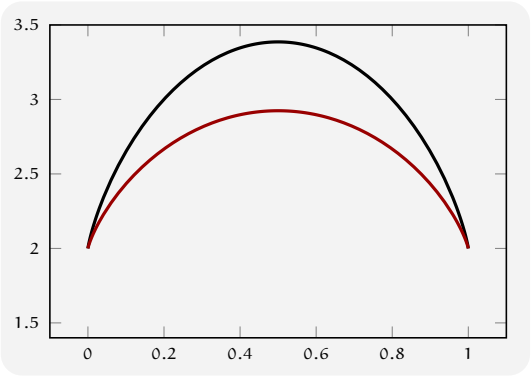
Analytic solutions for simplest cases

- **standard quickselect** $s(\alpha) = 2\,h(\alpha) + 2$

$$h(\alpha) = -\alpha \ln(\alpha) - (1-\alpha)\ln(1-\alpha)$$



YBB-quickselect $s(\alpha) = 1.\overline{3}\,h(\alpha) + 2$



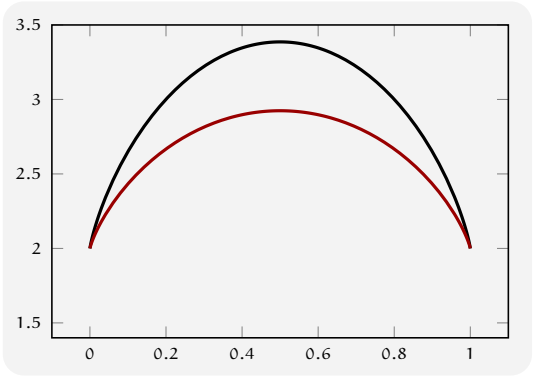
- **proportion from 2**



Linear-ranks Results



Analytic solutions for simplest cases



- **standard quickselect** $s(\alpha) = 2\,h(\alpha) + 2$

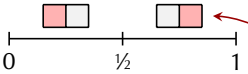
$$h(\alpha) = -\alpha \ln(\alpha) - (1-\alpha)\ln(1-\alpha)$$



- **YBB-quickselect** $s(\alpha) = 1.\overline{3}\,h(\alpha) + 2$

- **proportion from 2**

$$s(\alpha) = \begin{cases} -4.29394\alpha\ln(1-\alpha) + 4.29394\ln(1-\alpha) - 1.04899\alpha^3 - 3.14697\alpha^2 + 6.65176\alpha - 2\alpha\ln(\alpha) + 1.5 & \alpha \leqslant 0.5 \\ 2\alpha\ln(1-\alpha) - 2\ln(1-\alpha) + 1.04899\alpha^3 - 6.29394\alpha^2 + 2.78914\alpha + 4.29394\alpha\ln(\alpha) + 3.95581 & \alpha > 0.5 \end{cases}$$

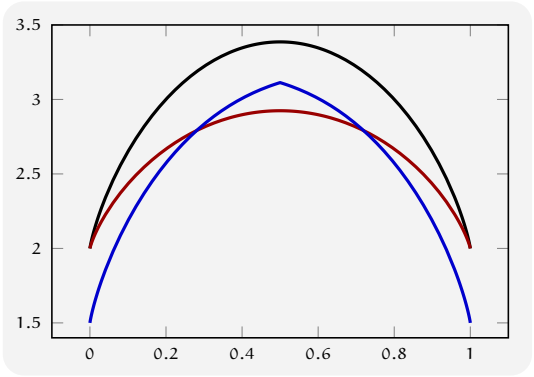


sample $k = 2$ elements;
use smaller if $\alpha \leqslant 0.5$,
use bigger otherwise

Linear-ranks Results



Analytic solutions for simplest cases



- **standard quickselect** $s(\alpha) = 2\,h(\alpha) + 2$

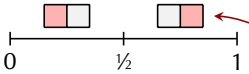
$$h(\alpha) = -\alpha \ln(\alpha) - (1-\alpha)\ln(1-\alpha)$$



YBB-quickselect $s(\alpha) = 1.\overline{3}\,h(\alpha) + 2$

- **proportion from 2**

$$s(\alpha) = \begin{cases} -4.29394\alpha\ln(1-\alpha) + 4.29394\ln(1-\alpha) - 1.04899\alpha^3 - 3.14697\alpha^2 + 6.65176\alpha - 2\alpha\ln(\alpha) + 1.5 & \alpha \leqslant 0.5 \\ 2\alpha\ln(1-\alpha) - 2\ln(1-\alpha) + 1.04899\alpha^3 - 6.29394\alpha^2 + 2.78914\alpha + 4.29394\alpha\ln(\alpha) + 3.95581 & \alpha > 0.5 \end{cases}$$

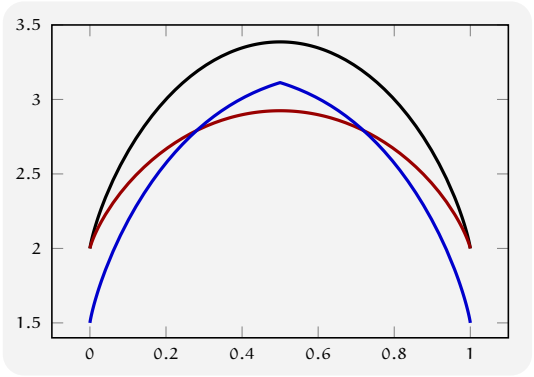


sample $k = 2$ elements;
use smaller if $\alpha \leqslant 0.5$,
use bigger otherwise

Linear-ranks Results



Analytic solutions for simplest cases



- **standard quickselect** $s(\alpha) = 2\,h(\alpha) + 2$

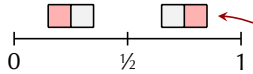
$$h(\alpha) = -\alpha \ln(\alpha) - (1-\alpha)\ln(1-\alpha)$$



YBB-quickselect $s(\alpha) = 1.\overline{3}\,h(\alpha) + 2$

- **proportion from 2**

$$s(\alpha) = \begin{cases} -4.29394\alpha\ln(1-\alpha) + 4.29394\ln(1-\alpha) - 1.04899\alpha^3 - 3.14697\alpha^2 + 6.65176\alpha - 2\alpha\ln(\alpha) + 1.5 & \alpha \leqslant 0.5 \\ 2\alpha\ln(1-\alpha) - 2\ln(1-\alpha) + 1.04899\alpha^3 - 6.29394\alpha^2 + 2.78914\alpha + 4.29394\alpha\ln(\alpha) + 3.95581 & \alpha > 0.5 \end{cases}$$



sample $k = 2$ elements;
use smaller if $\alpha \leqslant 0.5$,
use bigger otherwise

Can we get the best of both worlds?

Outline

1 Introduction

2 Random Ranks

3 Linear Ranks

4 Sesquickselect

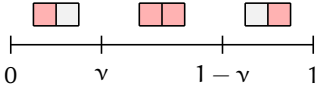
5 Conclusion

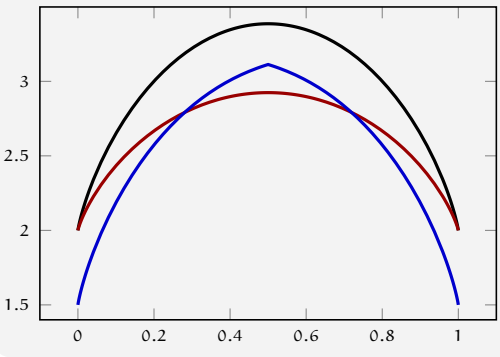
Sesqui-pivoting

⇝ *Use min, max **or** both as pivots*

"1.5 pivots"

Sesquickselect:



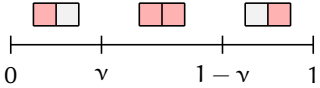


Sesqui-pivoting

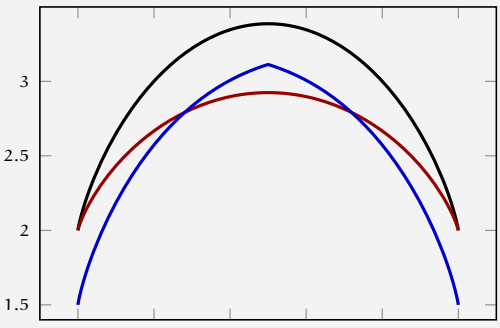
⇝ *Use min, max **or** both as pivots*

"1.5 pivots"

Sesquickselect:



- when to switch? $\nu \in (0, \frac{1}{2})$

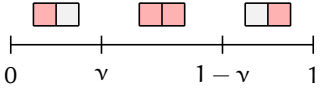


Sesqui-pivoting

⇝ *Use min, max **or** both as pivots*

"1.5 pivots"

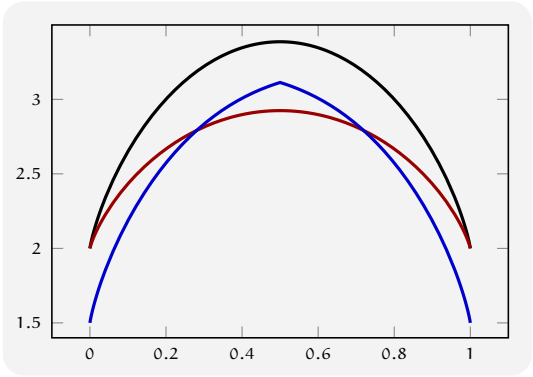
Sesquickselect:



- when to switch? $\nu \in (0, \frac{1}{2})$



solve $\int$-eq symbolically in ν

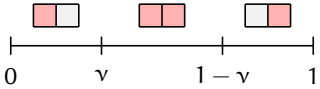


Sesqui-pivoting

⇝ *Use min, max **or** both as pivots*

"1.5 pivots"

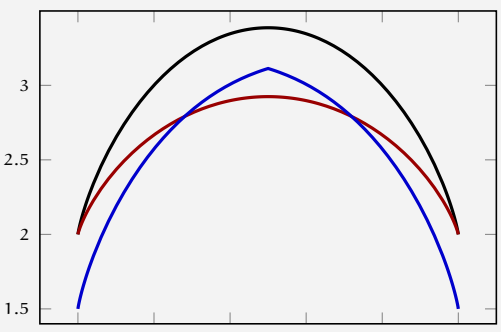
Sesquickselect:



- when to switch? $\nu \in (0, \frac{1}{2})$



solve $\int$-eq symbolically in ν



⇝ analytical proof: unique optimal $\nu = \nu^* \approx 0.265\,717$

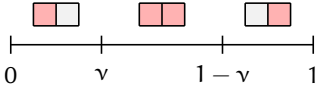
(*not* where red and blue lines cross …)

Sesqui-pivoting

⇝ *Use min, max **or** both as pivots*

"1.5 pivots"

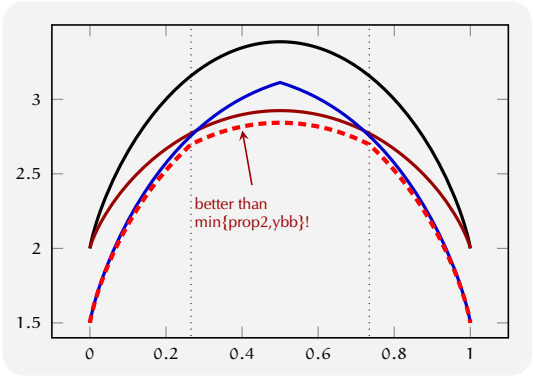
Sesquickselect:



- when to switch? $\nu \in (0, \frac{1}{2})$



solve $\int$-eq symbolically in ν



⇝ analytical proof: unique optimal $\nu = \nu^* \approx 0.265\,717$

(*not* where red and blue lines cross …)

$$s(\alpha) = \begin{cases} -4.63627\alpha^3 - 13.9088\alpha^2 + 27.9124\alpha - 25.8176\alpha\ln(1-\alpha) - 2\alpha\ln(\alpha) + 25.8176\ln(1-\alpha) + \frac{3}{2} & \alpha < \nu^* \\ 1.27709\alpha\ln(1-\alpha) - 1.27709\ln(1-\alpha) - 1.27709\alpha\ln(\alpha) + 1.95753 & \nu^* \leqslant \alpha \leqslant 1-\nu^* \\ 4.63627\alpha^3 - 27.8176\alpha^2 + 13.814\alpha + 2\alpha\ln(1-\alpha) + 25.8176\alpha\ln(\alpha) - 2\ln(1-\alpha) + 10.8674 & \alpha > \nu^* \end{cases}$$

Larger samples sizes

Sample size $k = 2$ *very small* ...

SQSk: Allow to choose 1 or 2 pivots out of k (depending on α)

code on github

- Solving differential equations seems intractable $\rightsquigarrow$ numeric approximations

Larger samples sizes

Sample size $k = 2$ *very small* ...

SQSk: Allow to choose 1 or 2 pivots out of k (depending on α)

code on github

- Solving differential equations seems intractable $\rightsquigarrow$ numeric approximations

Larger samples sizes

Sample size $k = 2$ *very small* ...

SQSk: Allow to choose 1 or 2 pivots out of k (depending on α)

code on github

- Solving differential equations seems intractable $\rightsquigarrow$ numeric approximations

Larger samples sizes

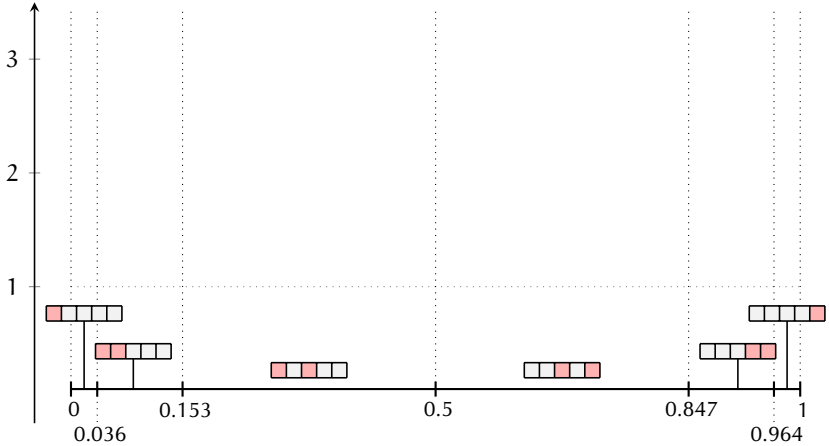
Sample size $k = 2$ *very small* ...

SQSk: Allow to choose 1 or 2 pivots out of k (depending on α)

code on github

- Solving differential equations seems intractable $\rightsquigarrow$ numeric approximations

Conjectured (approx) optimal for $k = 5$:



Larger samples sizes

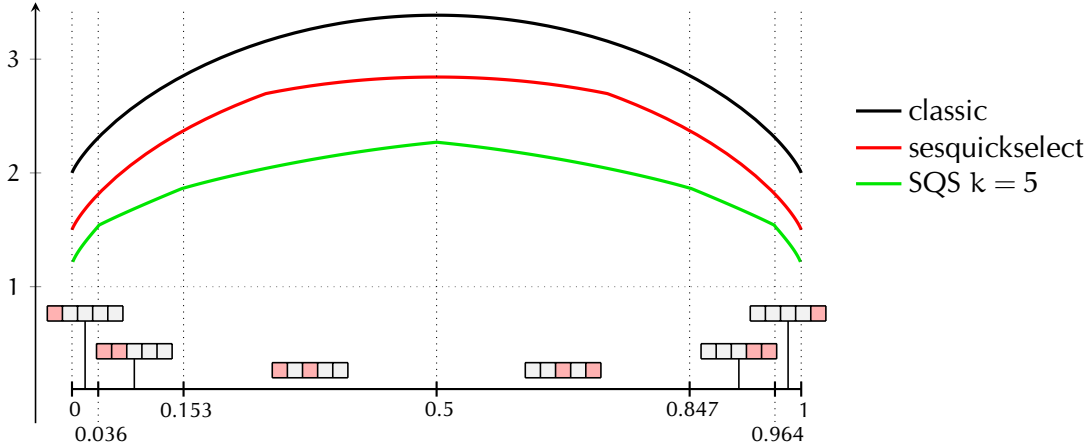
Sample size $k = 2$ *very small* ...

SQSk: Allow to choose 1 or 2 pivots out of k (depending on α)

code on github

- Solving differential equations seems intractable $\rightsquigarrow$ numeric approximations

Conjectured (approx) optimal for $k = 5$:



Larger samples sizes

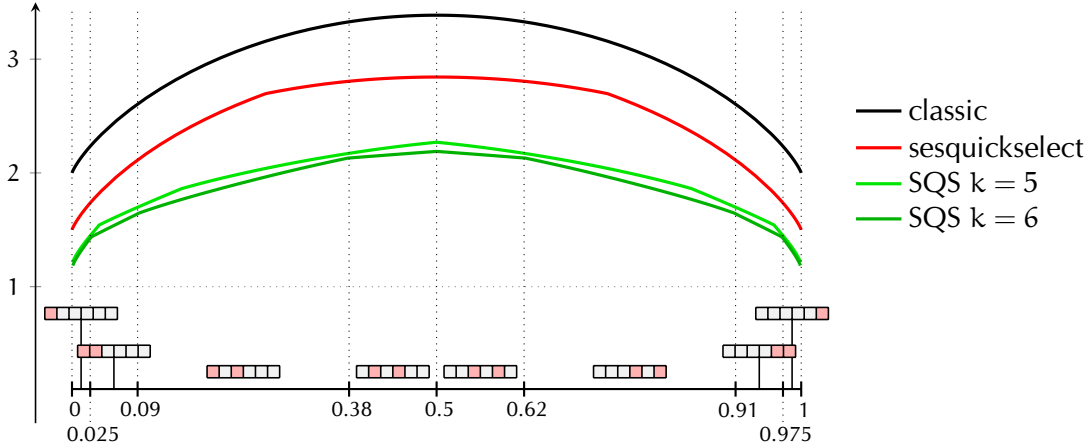
Sample size $k = 2$ *very small* ...

SQSk: Allow to choose 1 or 2 pivots out of k (depending on α)

code on github

- Solving differential equations seems intractable $\rightsquigarrow$ numeric approximations

Conjectured (approx) optimal for $k = 6$:



Larger samples sizes

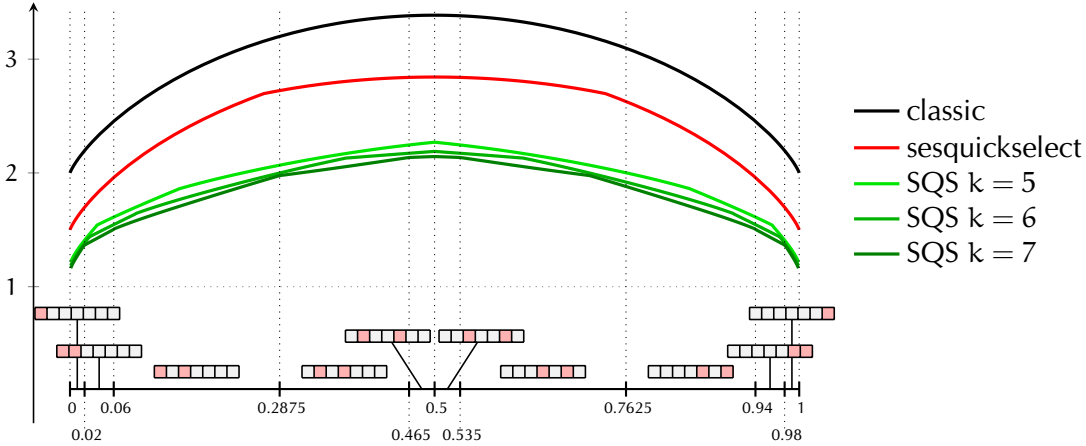
Sample size $k = 2$ *very small* ...

SQSk: Allow to choose 1 or 2 pivots out of k (depending on α)

code on github

- Solving differential equations seems intractable $\rightsquigarrow$ numeric approximations

Conjectured (approx) optimal for $k = 7$:



Outline

1 Introduction

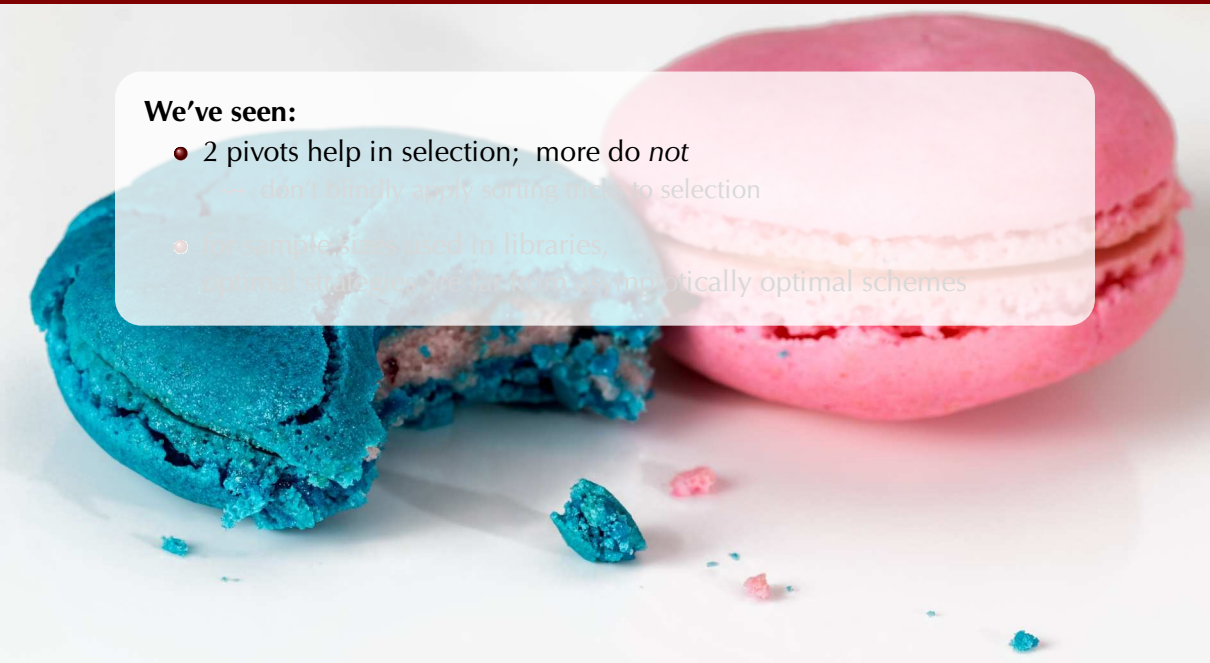
2 Random Ranks

3 Linear Ranks

4 Sesquickselect

5 Conclusion

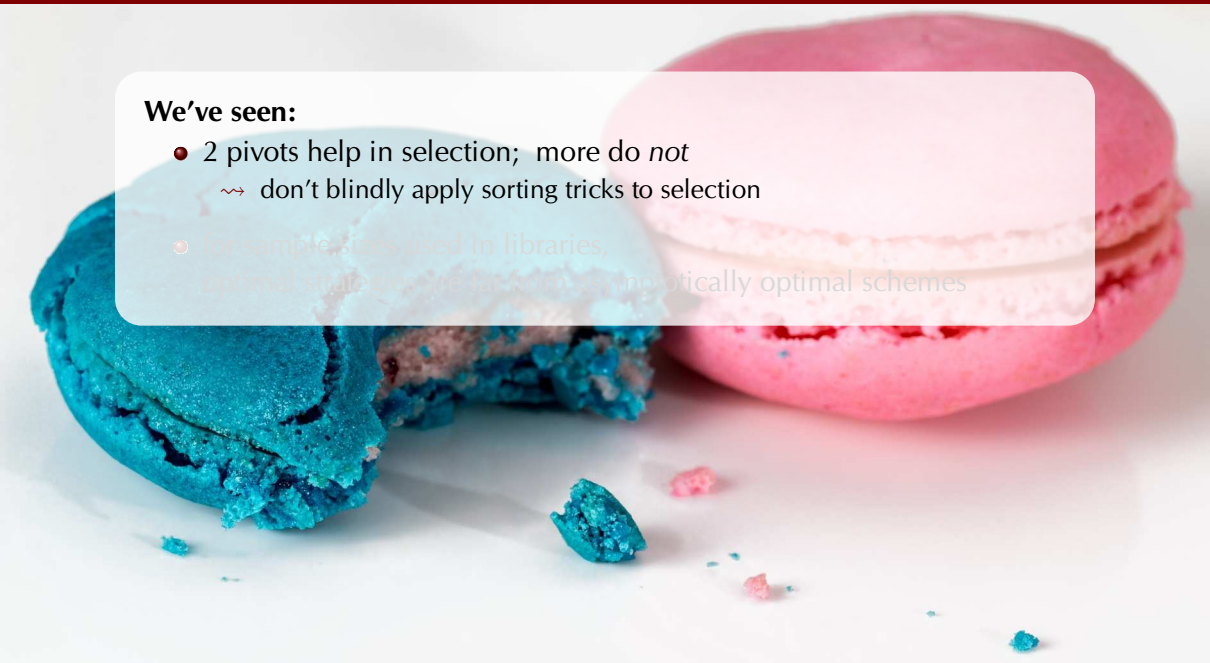
Conclusion



We've seen:

- 2 pivots help in selection; more do *not*
  ↝ don't blindly apply sorting tricks to selection
- for sample sizes used in libraries,
  optimal strategy is very far from asymptotically optimal schemes

Conclusion



We've seen:

- 2 pivots help in selection; more do *not*
  - ⇝ don't blindly apply sorting tricks to selection
- for sample sizes used in libraries, optimal strategy is very far from asymptotically optimal schemes

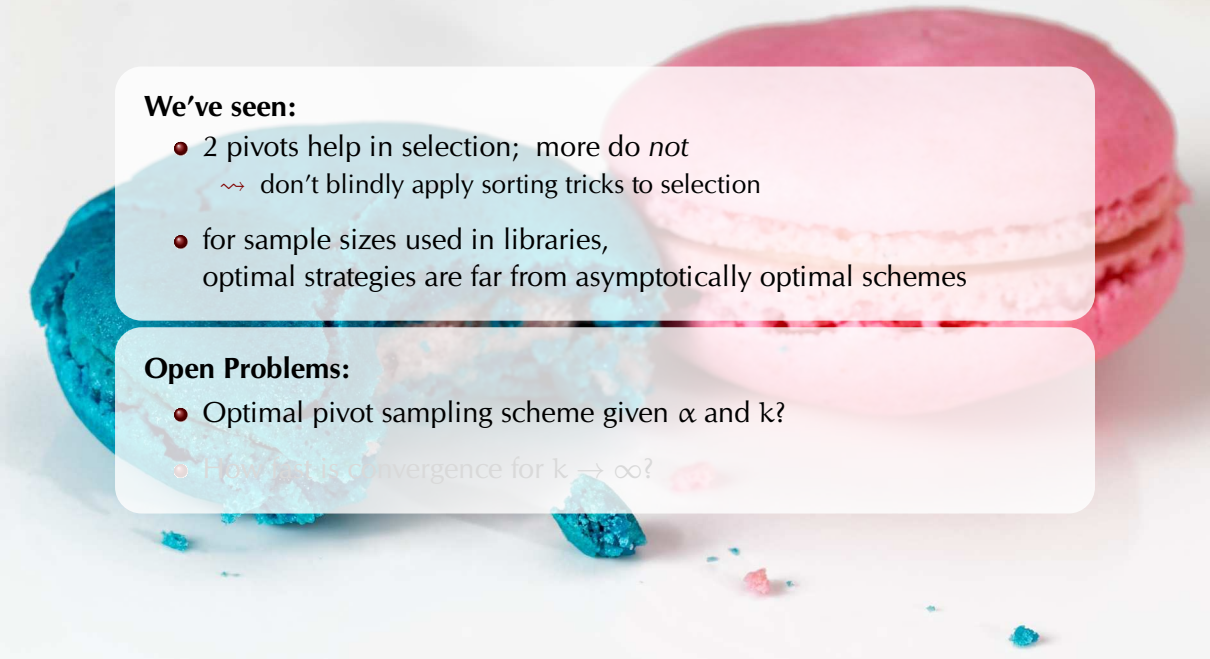
Conclusion



We've seen:

- 2 pivots help in selection; more do *not*
  - ⇝ don't blindly apply sorting tricks to selection
- for sample sizes used in libraries,
  optimal strategies are far from asymptotically optimal schemes

Conclusion



We've seen:

- 2 pivots help in selection; more do *not*
  - ⇝ don't blindly apply sorting tricks to selection
- for sample sizes used in libraries,
  optimal strategies are far from asymptotically optimal schemes

Open Problems:

- Optimal pivot sampling scheme given α and k?
- How fast is convergence for $k \to \infty$?

Conclusion

We've seen:

- 2 pivots help in selection; more do *not*
  - $\rightsquigarrow$ don't blindly apply sorting tricks to selection
- for sample sizes used in libraries,
  optimal strategies are far from asymptotically optimal schemes

Open Problems:

- Optimal pivot sampling scheme given α and k?
- How fast is convergence for $k \to \infty$?


KEEP
OFF
MEDIAN

Icons made by *Freepik*, *Gregor Cresnar*, *Those Icons*, *Smashicons*, *Good Ware*, *Pause08*, and *Madebyoliver* from `www.flaticon.com`.

3-way railway switch by *Asiano*
processor by *Konstantin Lanzet*
RAM module by *Utente:Sassospicco*
keep off median sign by *Michael Rivera*
Comic by *Randall Munroe*: xkcd 1602
Other photos from `www.pixabay.com`.