Pivot Sampling in Dual-Pivot Quicksort

Sebastian Wild Markus E. Nebel

[wild, nebel] @cs.uni-kl.de



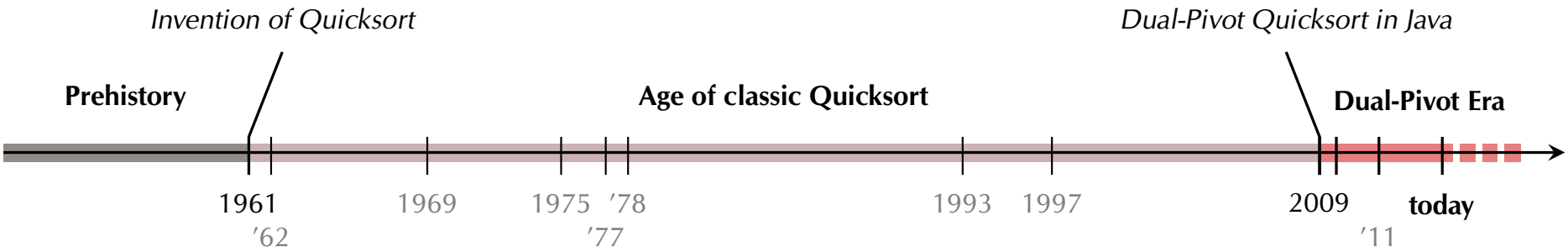
16 June 2014

25th International Conference on Probabilistic,
Combinatorial and Asymptotic Methods for the

Analysis of Algorithms

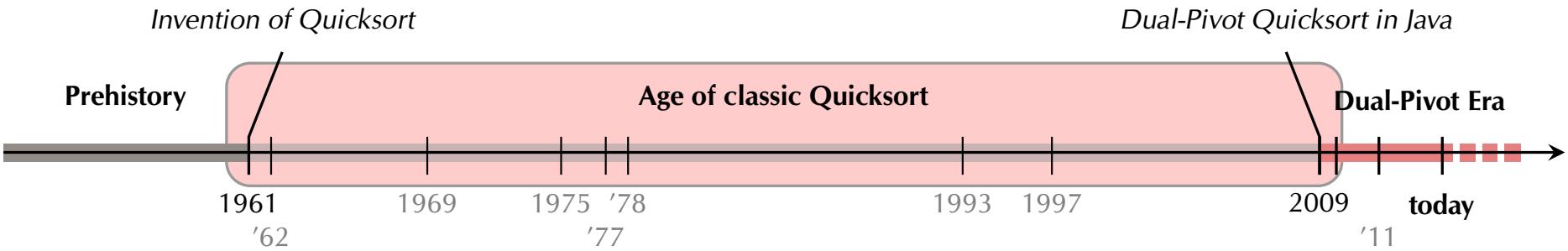
Sorting History

- Almost identical Quicksort in all programming libraries!
- **2009**: Java switches to **Yaroslavskiy's algorithm**
  - **Dual-Pivot** Quicksort with **new** partitioning method
  - Faster running time
  - No proper analysis at that time
- Why not earlier?
  - Other dual-pivot variants had been studied
  - Could **not** beat classic Quicksort



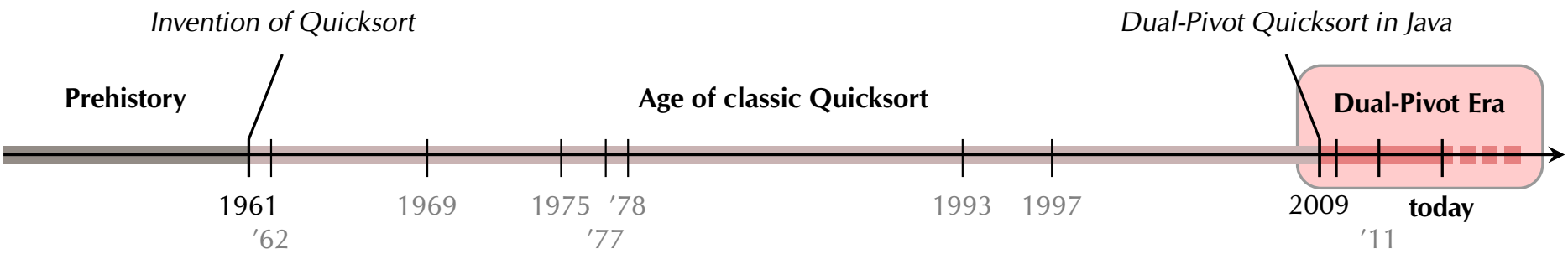
Sorting History

- Almost identical Quicksort in all programming libraries!
- **2009**: Java switches to **Yaroslavskiy's algorithm**
  - **Dual-Pivot** Quicksort with **new** partitioning method
  - Faster running time
  - No proper analysis at that time
- Why not earlier?
  - Other dual-pivot variants had been studied
  - Could **not** beat classic Quicksort



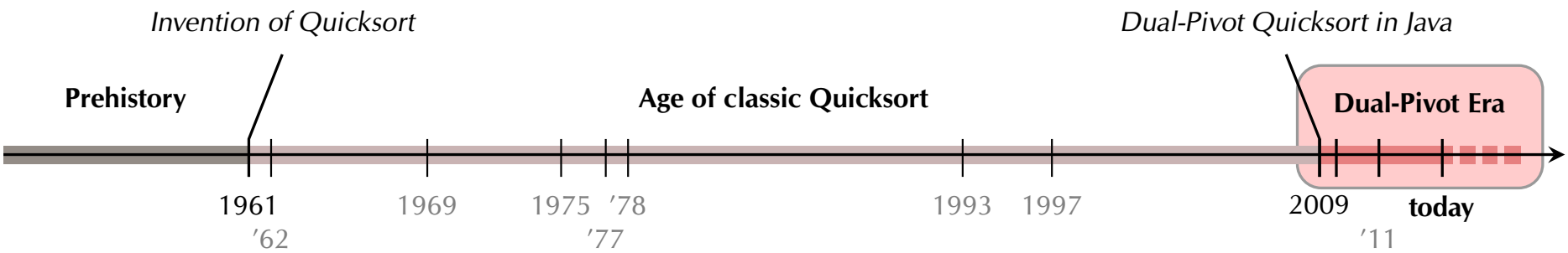
Sorting History

- Almost identical Quicksort in all programming libraries!
- **2009**: Java switches to **Yaroslavskiy's algorithm**
  - **Dual-Pivot** Quicksort with **new** partitioning method
  - Faster running time
  - No proper analysis at that time
- Why not earlier?
  - Other dual-pivot variants had been studied
  - Could **not** beat classic Quicksort



Sorting History

- Almost identical Quicksort in all programming libraries!
- **2009**: Java switches to **Yaroslavskiy's algorithm**
  - **Dual-Pivot** Quicksort with **new** partitioning method
  - Faster running time
  - No proper analysis at that time
- Why not earlier?
  - Other dual-pivot variants had been studied
  - Could **not** beat classic Quicksort



Previous Results (Average Case)

	Classic	Dual-Pivot	Relative
Running Time	(from various experiments)		−10±2%
Comparisons	$2\,n\ln n$	$1.9\,n\ln n$	−5%
Swaps	$0.\overline{3}\,n\ln n$	$0.6\,n\ln n$	+80%
Bytecode Instructions	$18\,n\ln n$	$21.7\,n\ln n$	+20.6%
MMIX oops υ	$11\,n\ln n$	$13.1\,n\ln n$	+19.1%
MMIX mems μ	$2.\overline{6}\,n\ln n$	$2.8\,n\ln n$	+5%
fresh elements ($\approx$ cache misses)	$2\,n\ln n$	$1.6\,n\ln n$	−20%

$+\,O(n)$, *average case results*

Results for pivots at fixed positions.

Previous Results (Average Case)

	Classic	Dual-Pivot	Relative
Running Time	(from various experiments)		−10±2%
Comparisons	$2\,n\ln n$	$1.9\,n\ln n$	−5%
Swaps	$0.\overline{3}\,n\ln n$	$0.6\,n\ln n$	+80%
Bytecode Instructions	$18\,n\ln n$	$21.7\,n\ln n$	+20.6%
MMIX oops υ	$11\,n\ln n$	$13.1\,n\ln n$	+19.1%
MMIX mems μ	$2.\overline{6}\,n\ln n$	$2.8\,n\ln n$	+5%
fresh elements ($\approx$ cache misses)	$2\,n\ln n$	$1.6\,n\ln n$	−20%

$+\,O(n)$, *average case results*

Results for pivots at fixed positions.

Previous Results (Average Case)

	Classic	**Dual-Pivot**	Relative
Running Time	(from various experiments)		$-10\pm2\%$
Comparisons	$2\,n\ln n$	$1.9\,n\ln n$	-5%
Swaps	$0.\overline{3}\,n\ln n$	$0.6\,n\ln n$	$+80\%$
Bytecode Instructions	$18\,n\ln n$	$21.7\,n\ln n$	$+20.6\%$
MMIX oops υ	$11\,n\ln n$	$13.1\,n\ln n$	$+19.1\%$
MMIX mems μ	$2.\overline{6}\,n\ln n$	$2.8\,n\ln n$	$+5\%$
fresh elements ($\approx$ cache misses)	$2\,n\ln n$	$1.6\,n\ln n$	-20%

$+\,O(n)$, *average case results*

Results for pivots at fixed positions.

Previous Results (Average Case)

	Classic	**Dual-Pivot**	Relative
Running Time	(from various experiments)		$-10\pm2\%$
Comparisons	$2\,n\ln n$	$1.9\,n\ln n$	-5%
Swaps	$0.\overline{3}\,n\ln n$	$0.6\,n\ln n$	$+80\%$
Bytecode Instructions	$18\,n\ln n$	$21.7\,n\ln n$	$+20.6\%$
MMIX oops υ	$11\,n\ln n$	$13.1\,n\ln n$	$+19.1\%$
MMIX mems μ	$2.\overline{6}\,n\ln n$	$2.8\,n\ln n$	$+5\%$
fresh elements ($\approx$ cache misses)	$2\,n\ln n$	$1.6\,n\ln n$	-20%

$+\,O(n)$, *average case results*

Results for pivots at fixed positions.

Previous Results (Average Case)

	Classic	**Dual-Pivot**	Relative
Running Time	(from various experiments)		$-10\pm2\%$
Comparisons	$2\,n\ln n$	$1.9\,n\ln n$	-5%
Swaps	$0.\overline{3}\,n\ln n$	$0.6\,n\ln n$	$+80\%$
Bytecode Instructions	$18\,n\ln n$	$21.7\,n\ln n$	$+20.6\%$
MMIX oops υ	$11\,n\ln n$	$13.1\,n\ln n$	$+19.1\%$
MMIX mems μ	$2.\overline{6}\,n\ln n$	$2.8\,n\ln n$	$+5\%$
fresh elements ($\approx$ cache misses)	$2\,n\ln n$	$1.6\,n\ln n$	-20%

$+\,O(n)$, *average case results*

Results for pivots at fixed positions. . . . nobody uses that!

⇝ **Here:** Choosing pivots from **sample**.

Previous Results (Average Case)

	Classic	**Dual-Pivot**	Relative
Running Time	(from various experiments)		$-10\pm2\%$
Comparisons	$2\,n\ln n$	$1.9\,n\ln n$	-5%
Swaps	$0.\overline{3}\,n\ln n$	$0.6\,n\ln n$	$+80\%$
Bytecode Instructions	$18\,n\ln n$	$21.7\,n\ln n$	$+20.6\%$
MMIX oops υ	$11\,n\ln n$	$13.1\,n\ln n$	$+19.1\%$
MMIX mems μ	$2.\overline{6}\,n\ln n$	$2.8\,n\ln n$	$+5\%$
fresh elements ($\approx$ cache misses)	$2\,n\ln n$	$1.6\,n\ln n$	-20%

$+\,O(n)$, *average case results*

Results for pivots at fixed positions. . . . nobody uses that!

$\rightsquigarrow$ **Here:** Choosing pivots from **sample**.

Dual Pivot Quicksort

Algorithm (Conceptual View)

1. Choose **two pivots** $P \leqslant Q$
2. For each element x, determine its **class**
   - **small** for $x < P$
   - **medium** for $P < x < Q$
   - **large** for $Q < x$

   by comparing x to **pivots** P and Q
3. Arrange elements according to classes:

4. Sort subarrays recursively.

How to implement 3 *efficiently on arrays?*

Dual Pivot Quicksort

Algorithm (Conceptual View)

1. Choose **two pivots** $P \leqslant Q$
2. For each element x, determine its **class**
   - **small** for $x < P$
   - **medium** for $P < x < Q$
   - **large** for $Q < x$

   by comparing x to **pivots** P and Q
3. Arrange elements according to classes:



4. Sort subarrays recursively.

How to implement 3 *efficiently on arrays?*

Dual Pivot Quicksort

Algorithm (Conceptual View)

1. Choose **two pivots** $P \leqslant Q$
2. For each element x, determine its **class**
   - **small** for $x < P$
   - **medium** for $P < x < Q$
   - **large** for $Q < x$

   by comparing x to **pivots** P and Q
3. Arrange elements according to classes:



4. Sort subarrays recursively.

How to implement 3 efficiently on arrays?

Dual Pivot Quicksort

Algorithm (Conceptual View)

1. Choose **two pivots** $P \leqslant Q$
2. For each element x, determine its **class**
   - **small** for $x < P$
   - **medium** for $P < x < Q$
   - **large** for $Q < x$

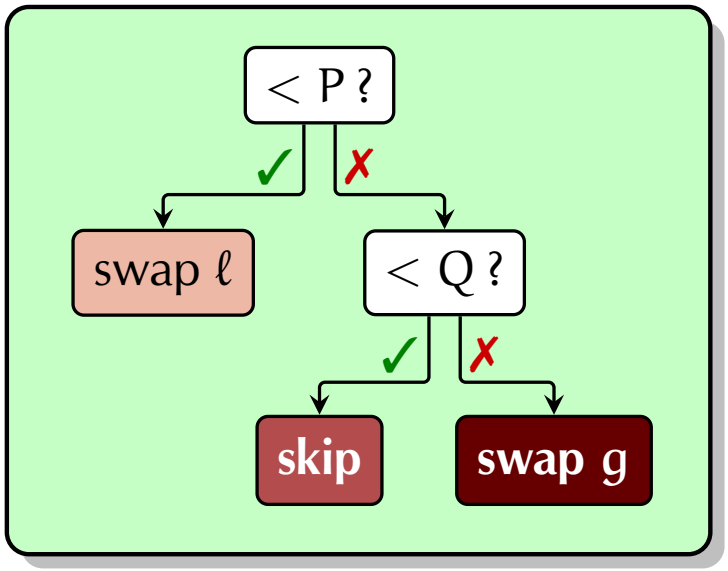
   by comparing x to **pivots** P and Q
3. Arrange elements according to classes:

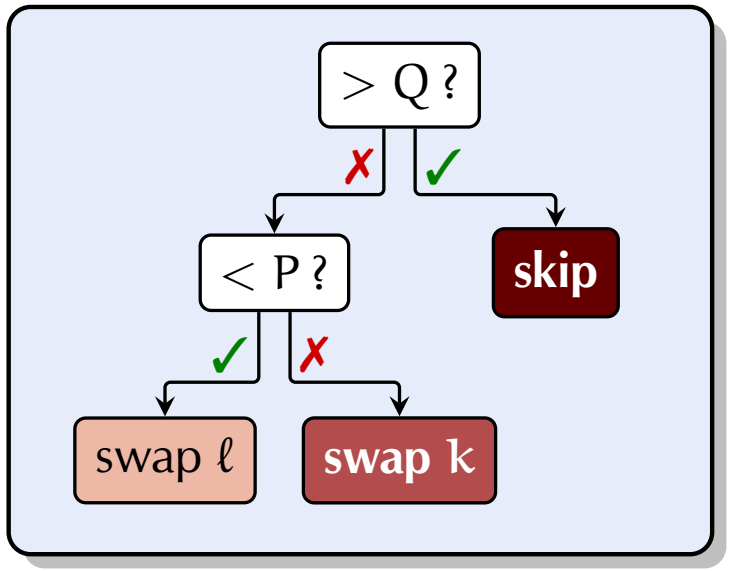


4. Sort subarrays recursively.

How to implement 3 *efficiently on arrays?*

Yaroslavskiy's Algorithm



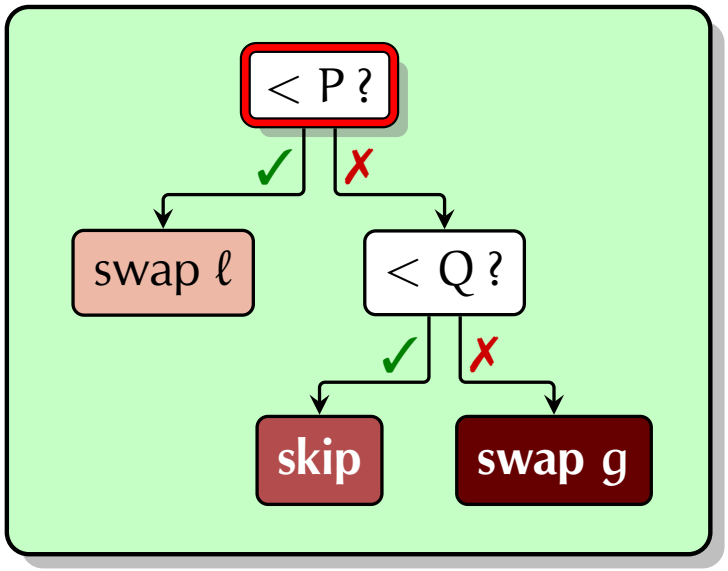


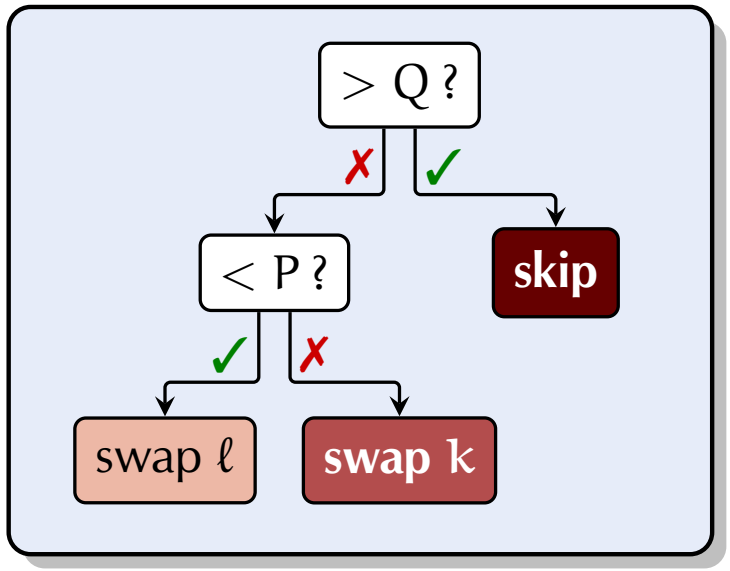


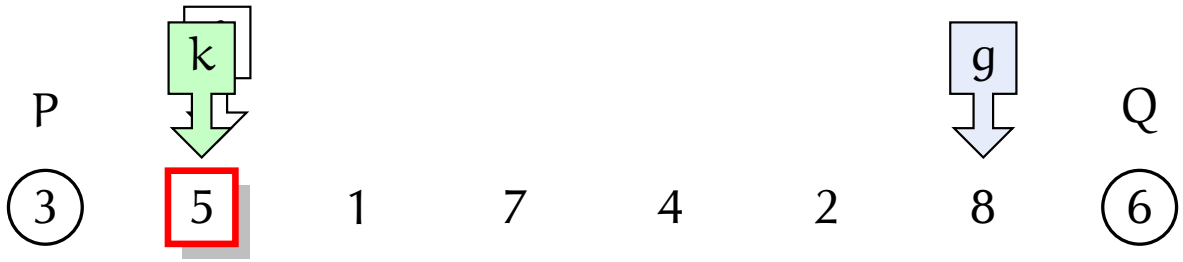
Invariant:



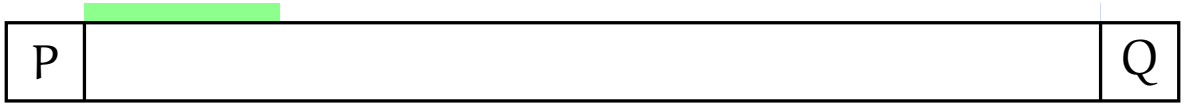
Yaroslavskiy's Algorithm



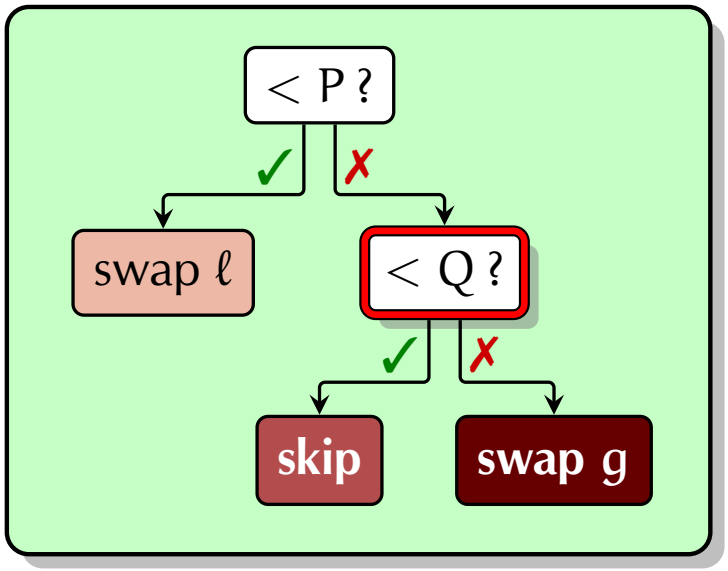


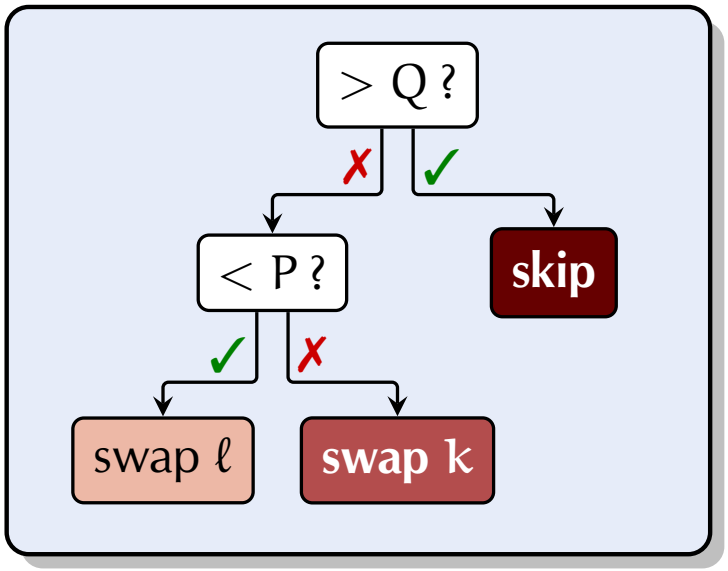


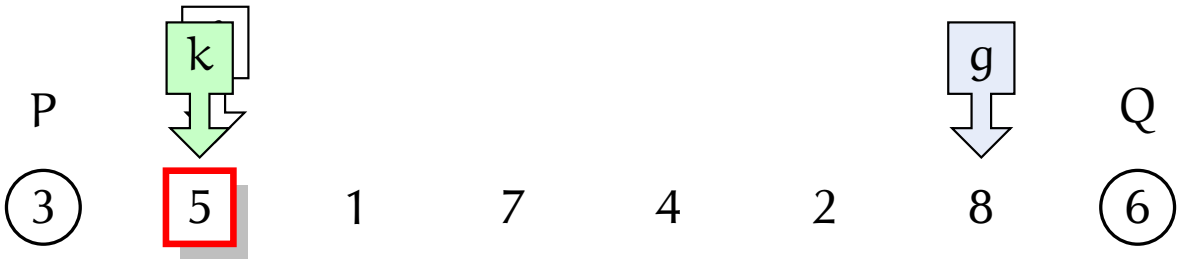
Invariant:



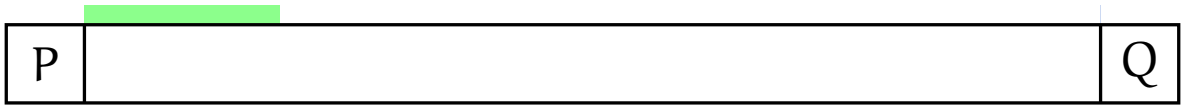
Yaroslavskiy's Algorithm



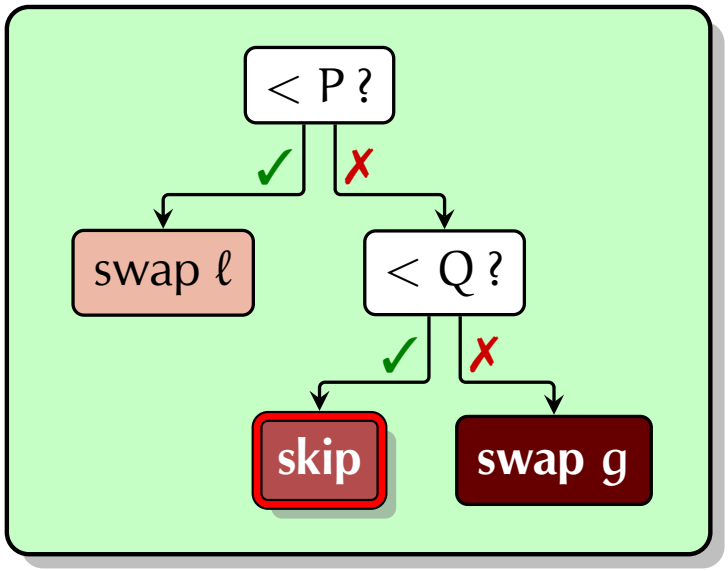


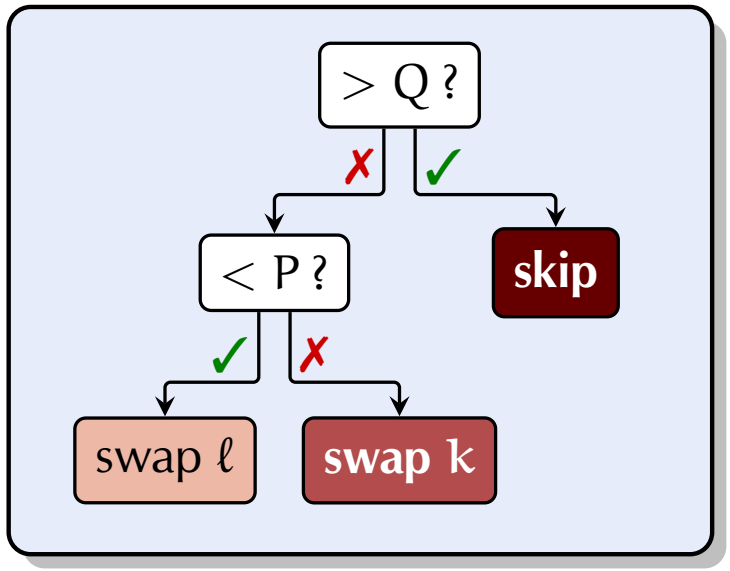


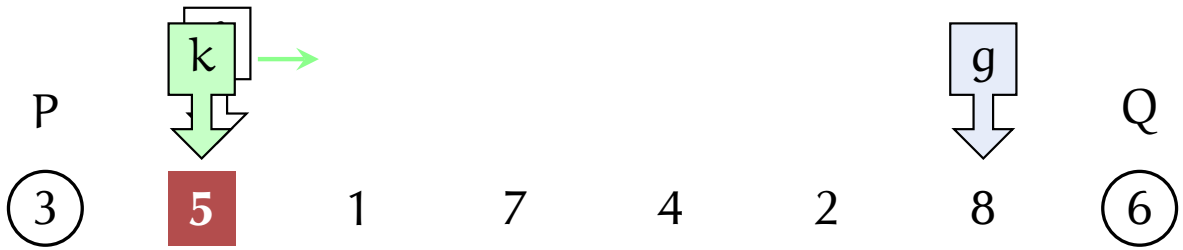
Invariant:



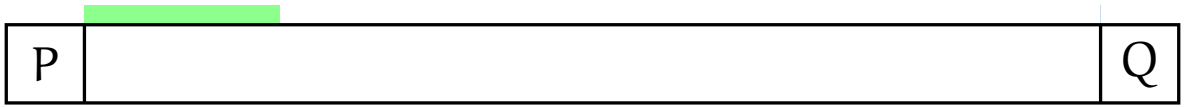
Yaroslavskiy's Algorithm



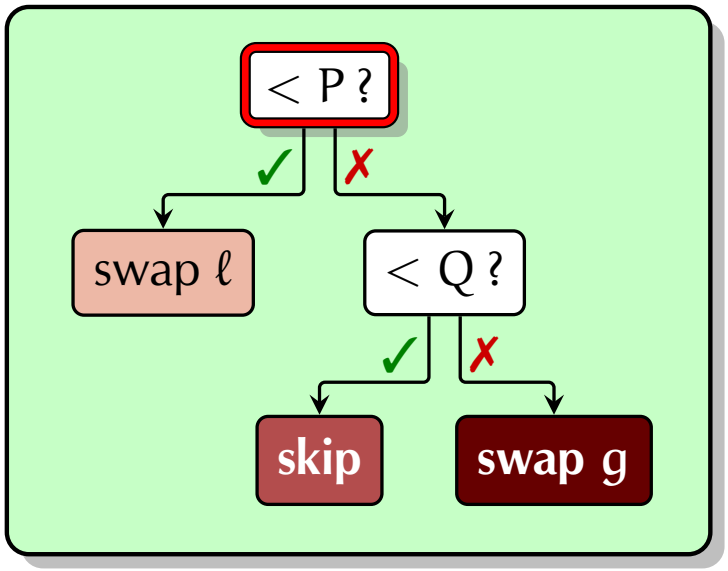


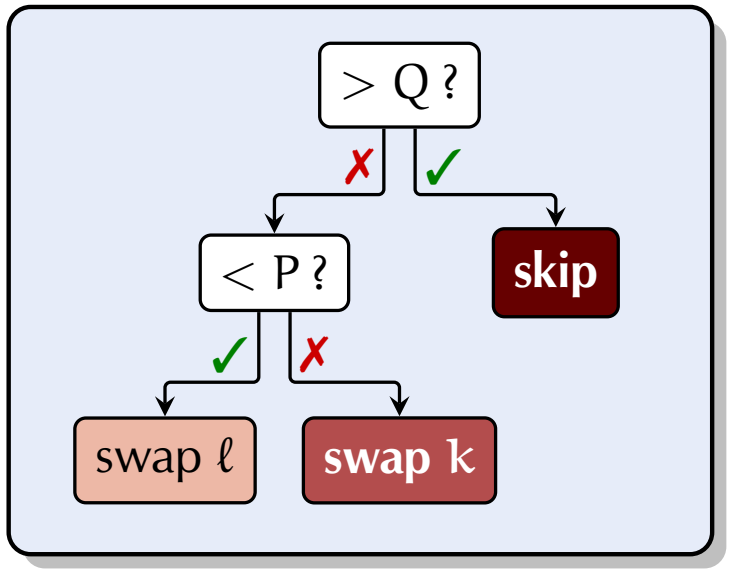


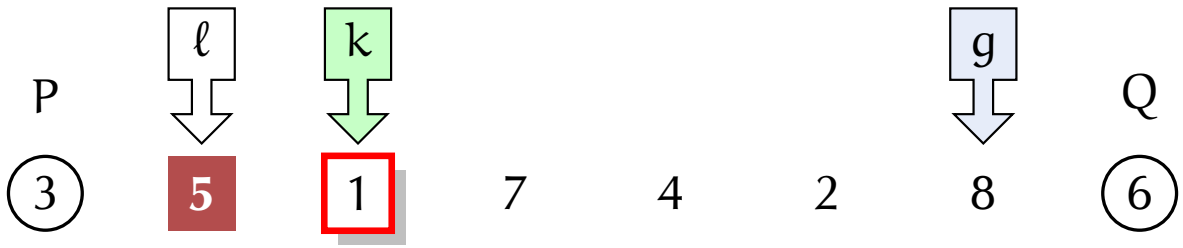
Invariant:



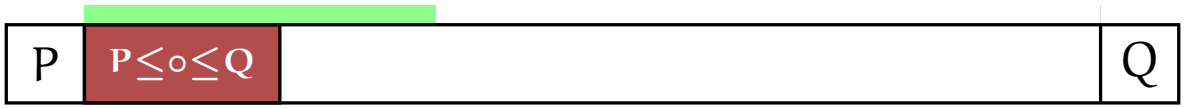
Yaroslavskiy's Algorithm



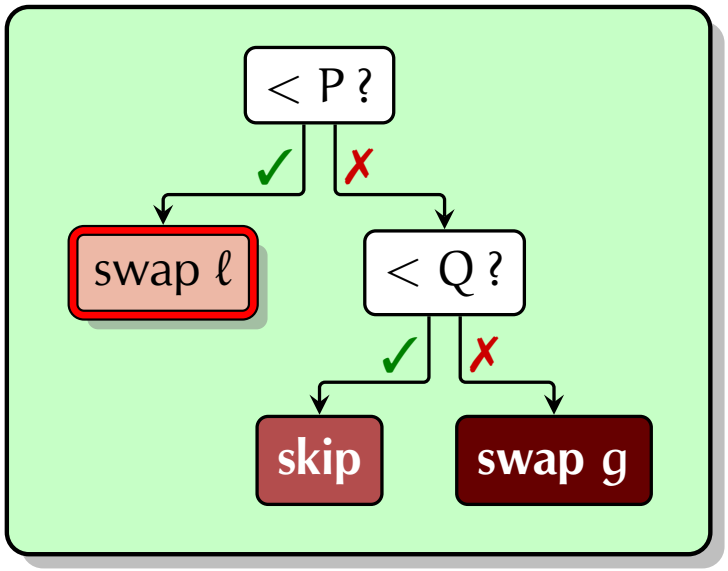


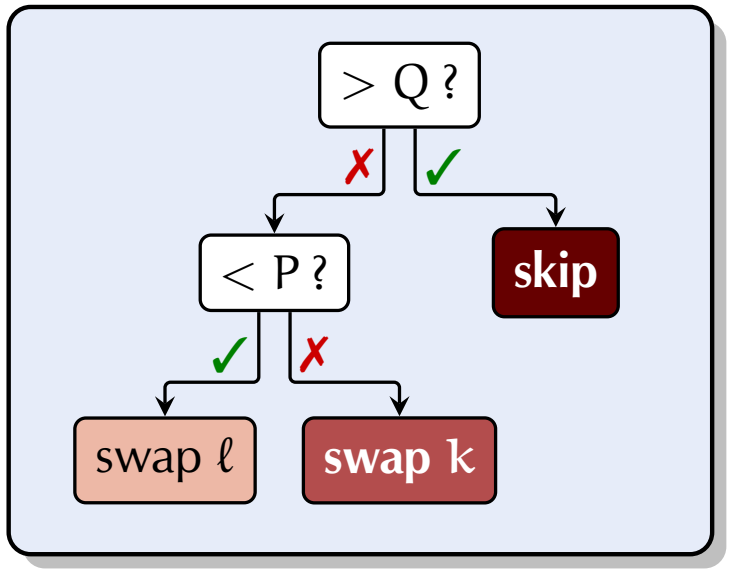


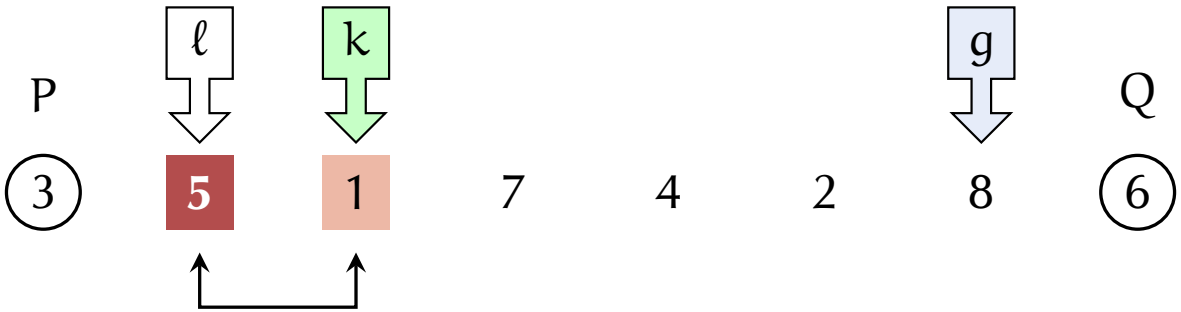
Invariant:



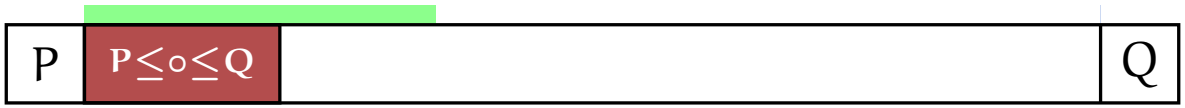
Yaroslavskiy's Algorithm



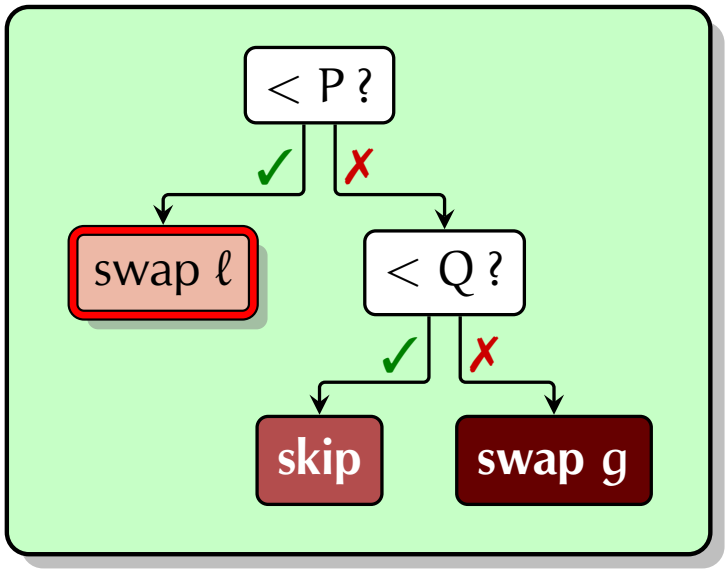


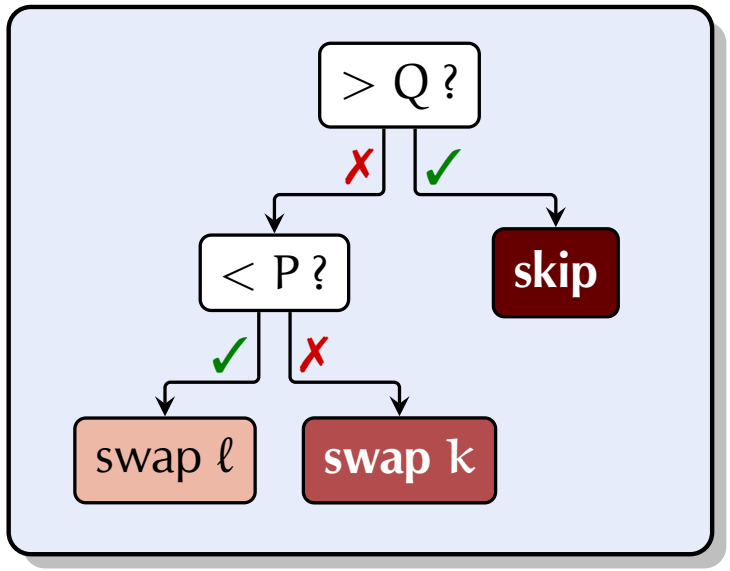


Invariant:



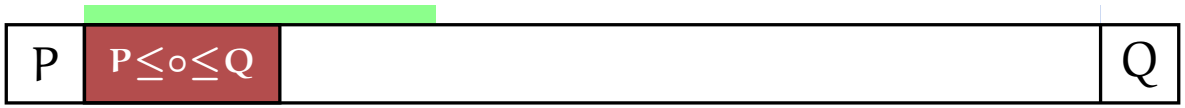
Yaroslavskiy's Algorithm



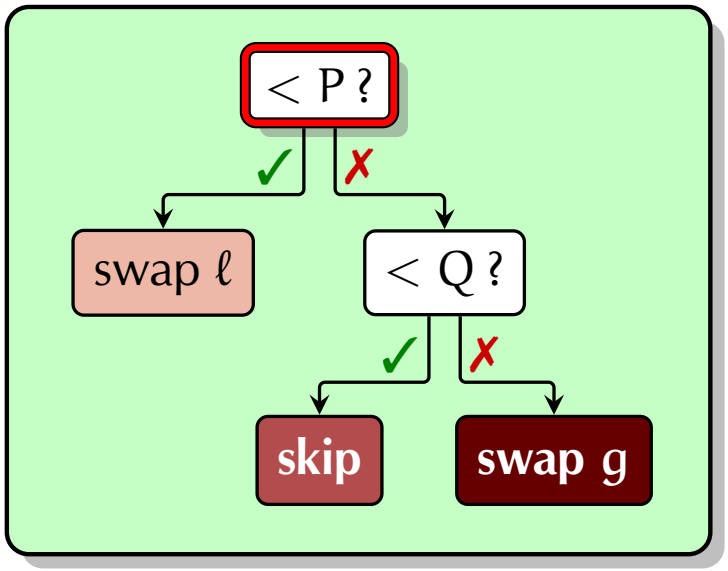


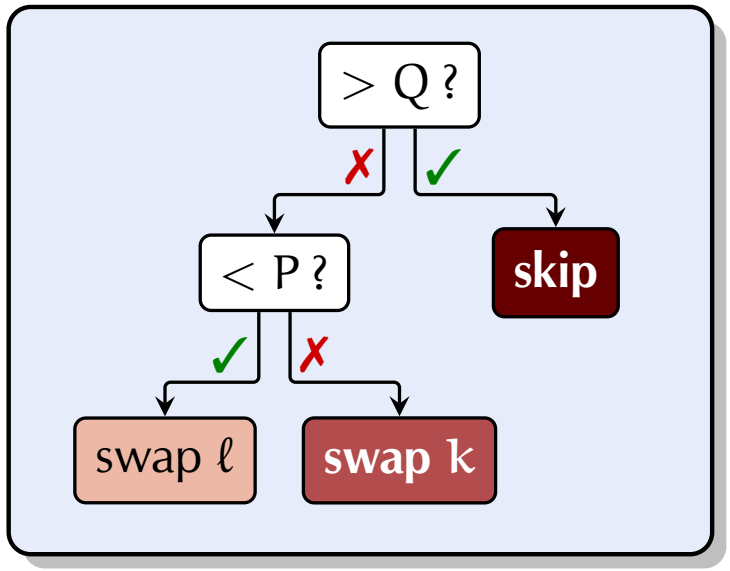


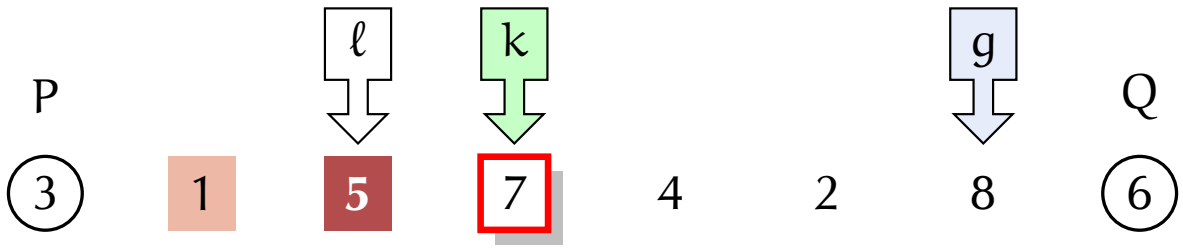
Invariant:



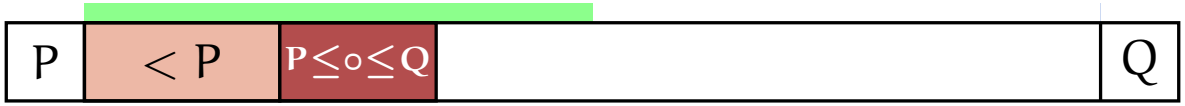
Yaroslavskiy's Algorithm



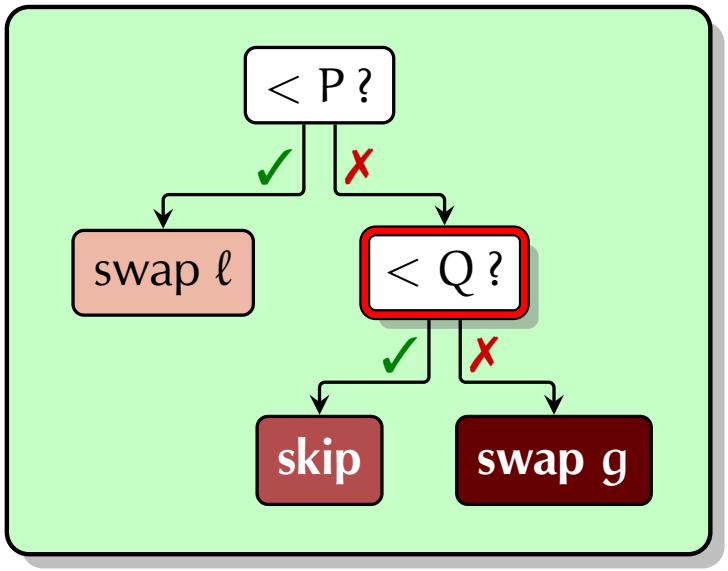


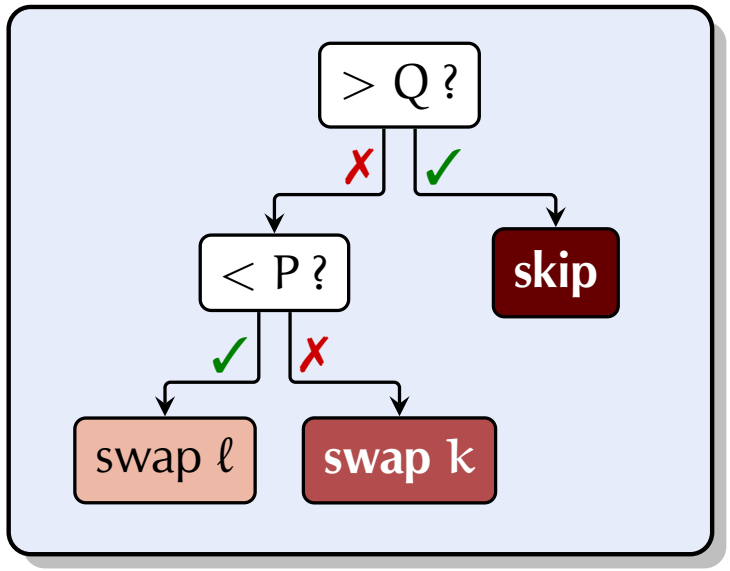


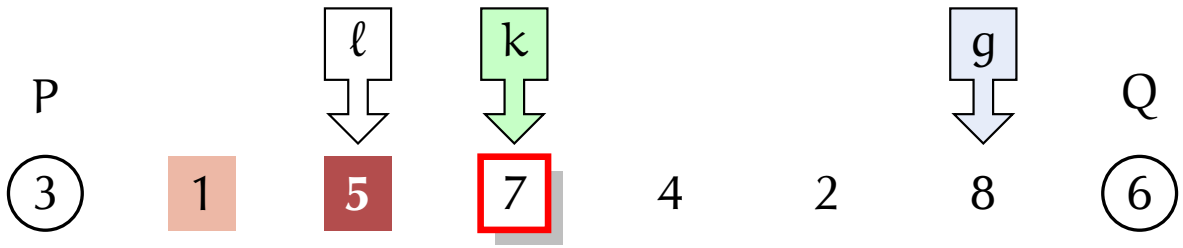
Invariant:



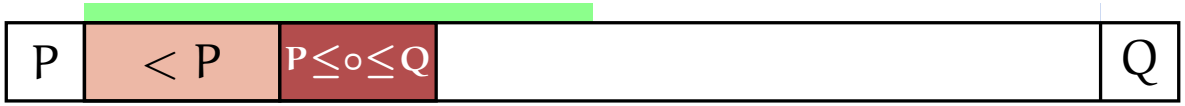
Yaroslavskiy's Algorithm



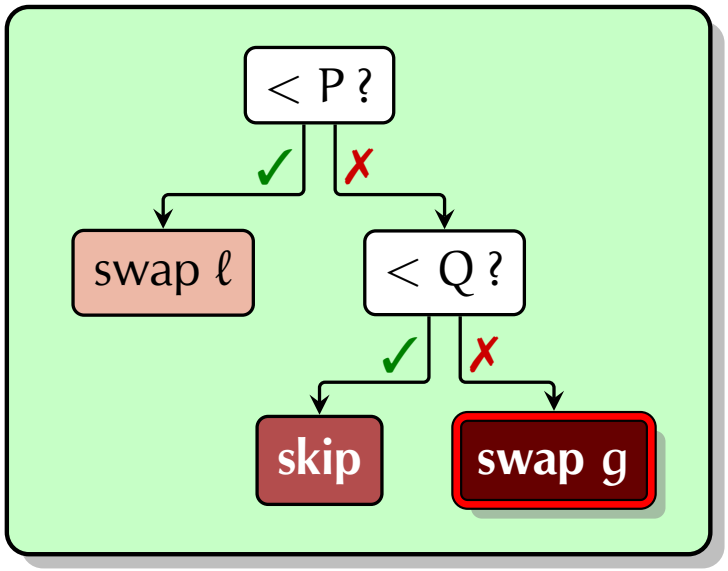


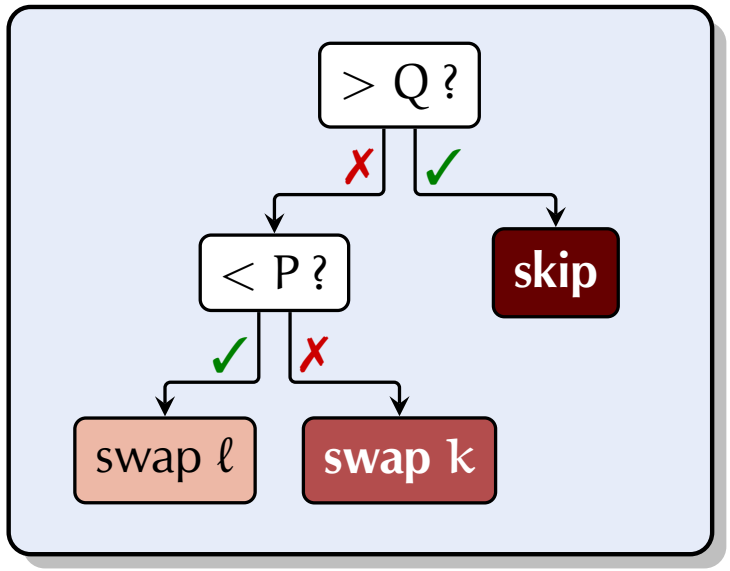


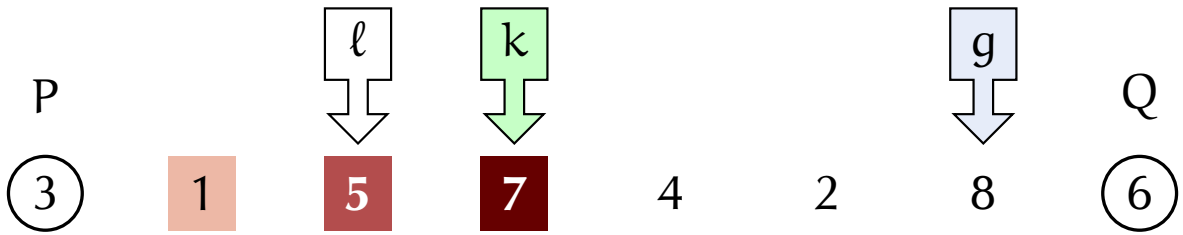
Invariant:



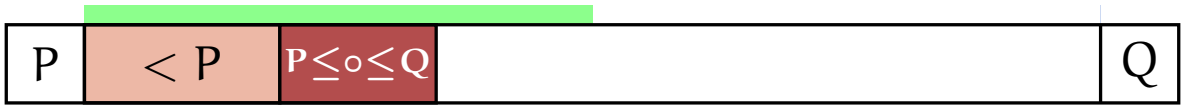
Yaroslavskiy's Algorithm



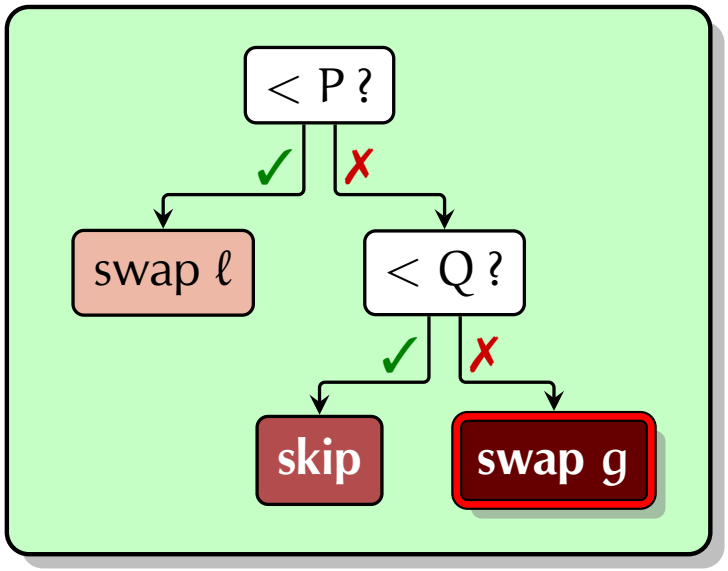


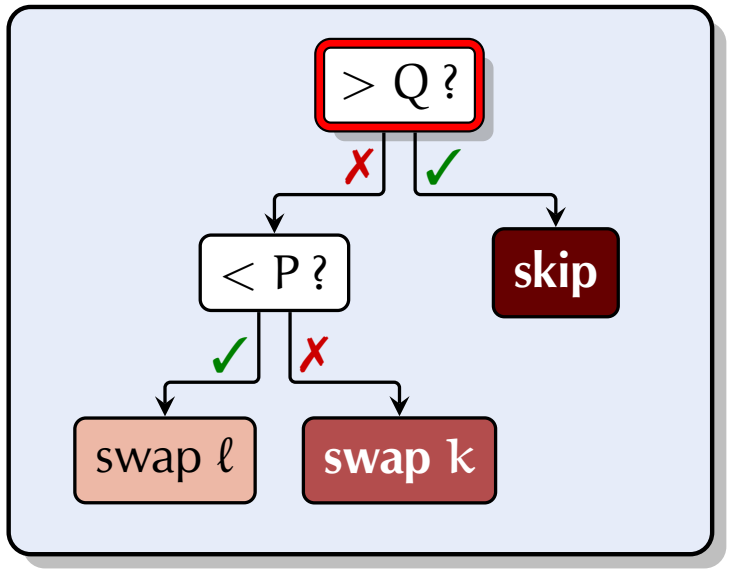


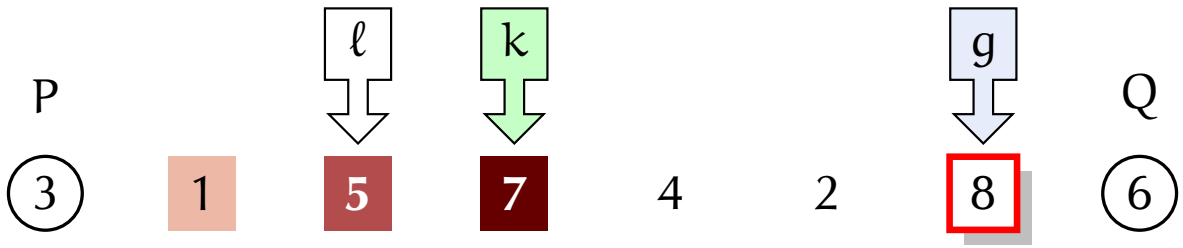
Invariant:



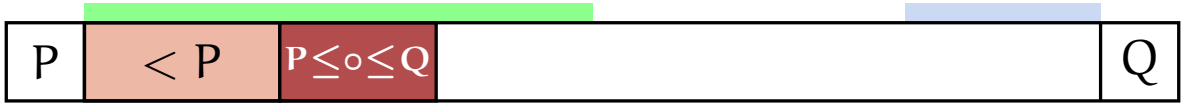
Yaroslavskiy's Algorithm



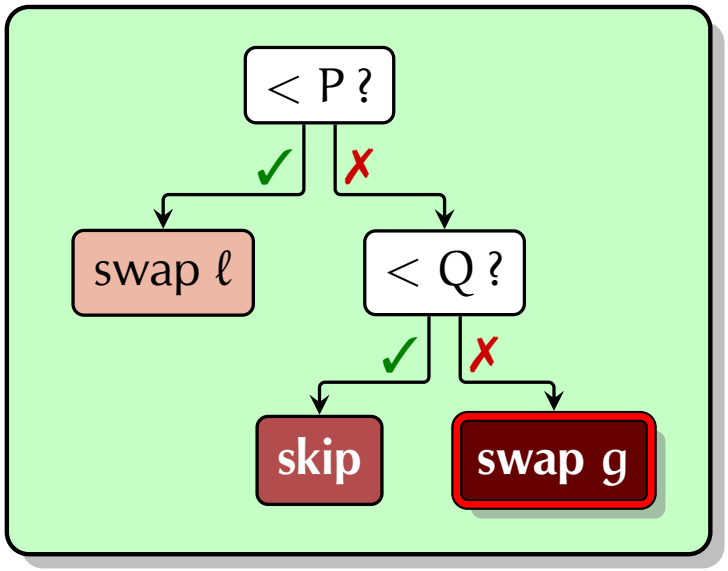


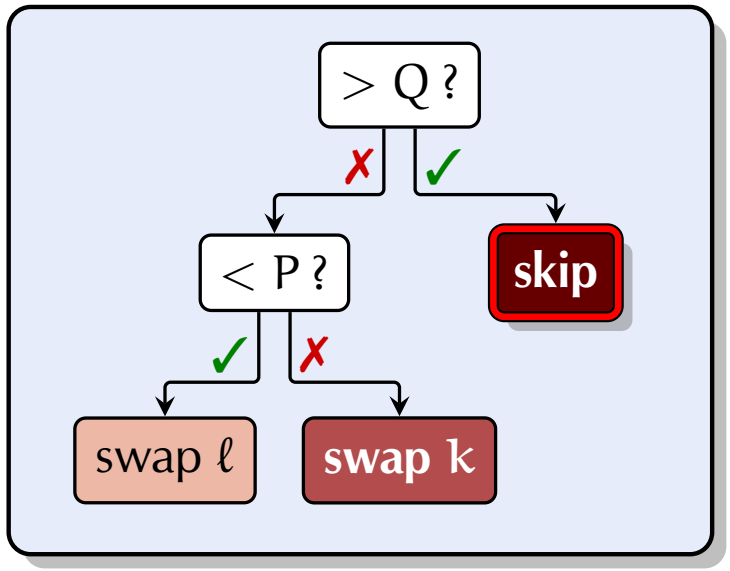


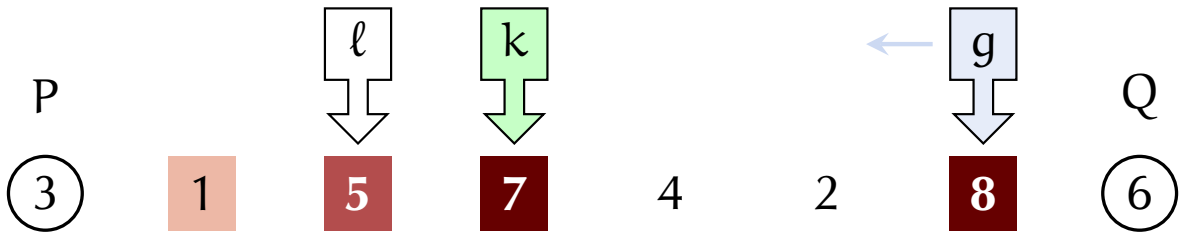
Invariant:



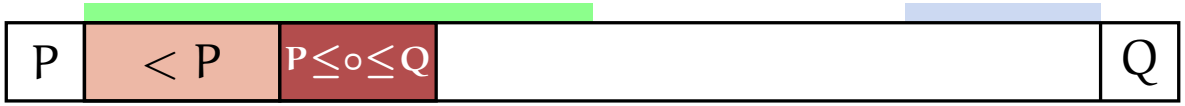
Yaroslavskiy's Algorithm



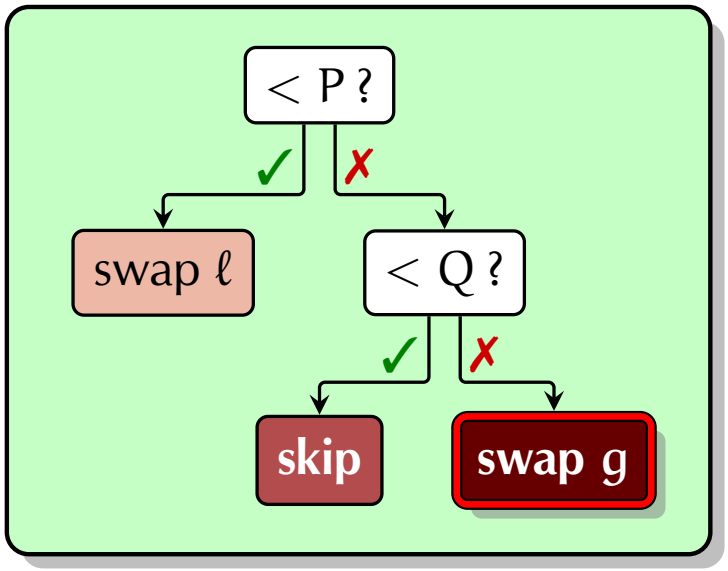


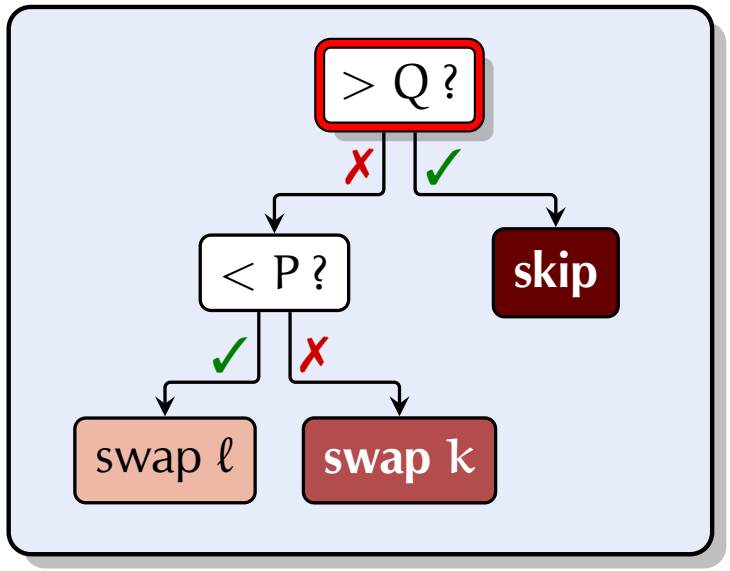


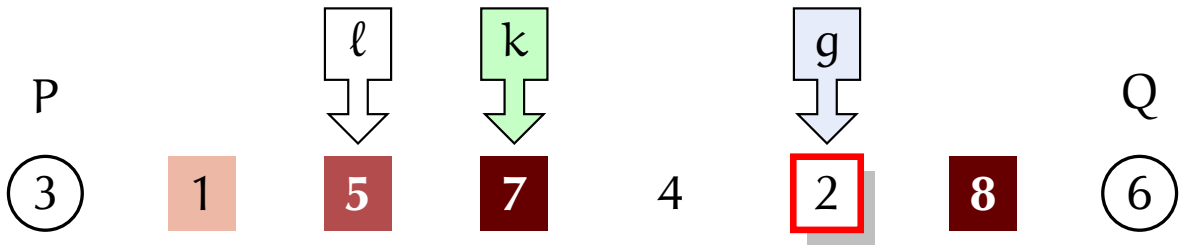
Invariant:



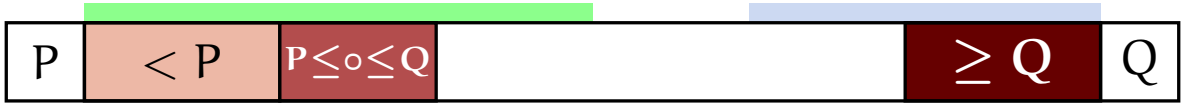
Yaroslavskiy's Algorithm



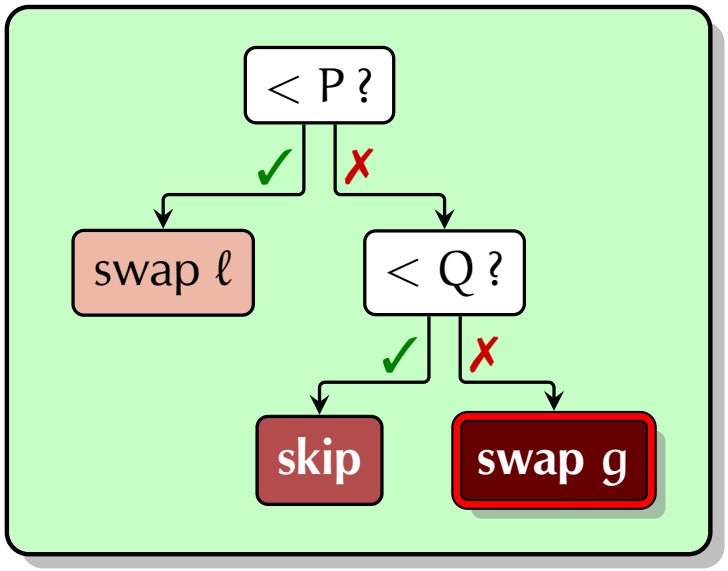


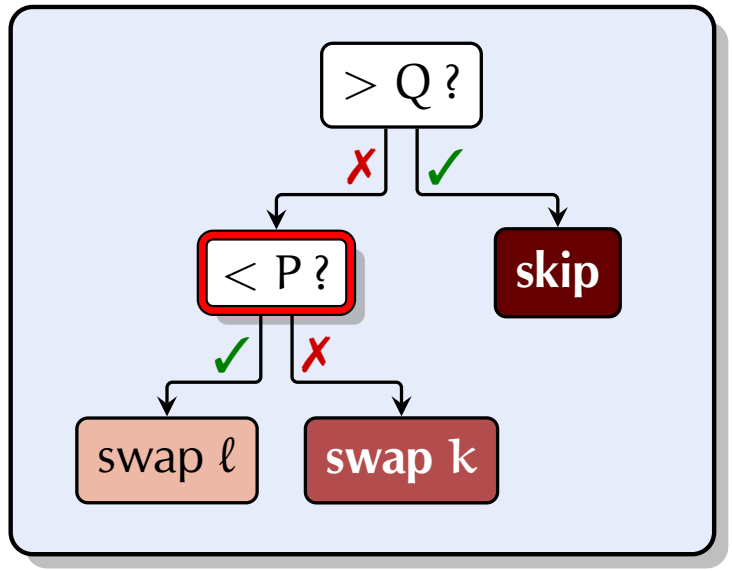


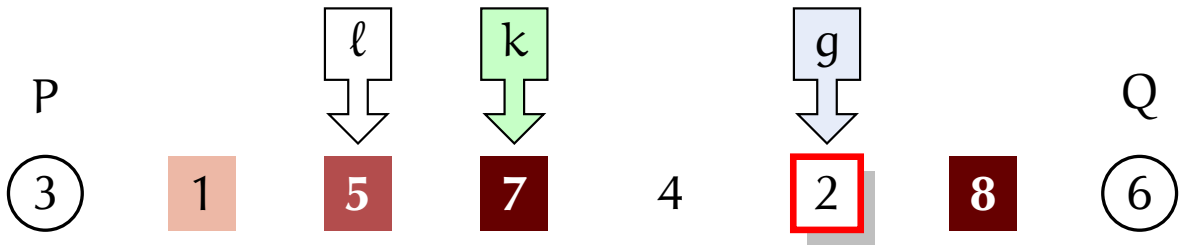
Invariant:



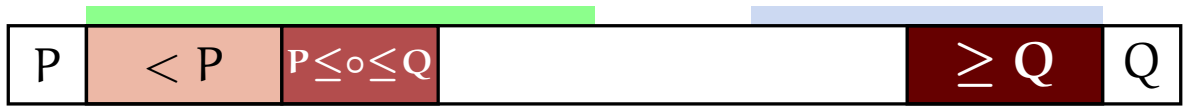
Yaroslavskiy's Algorithm



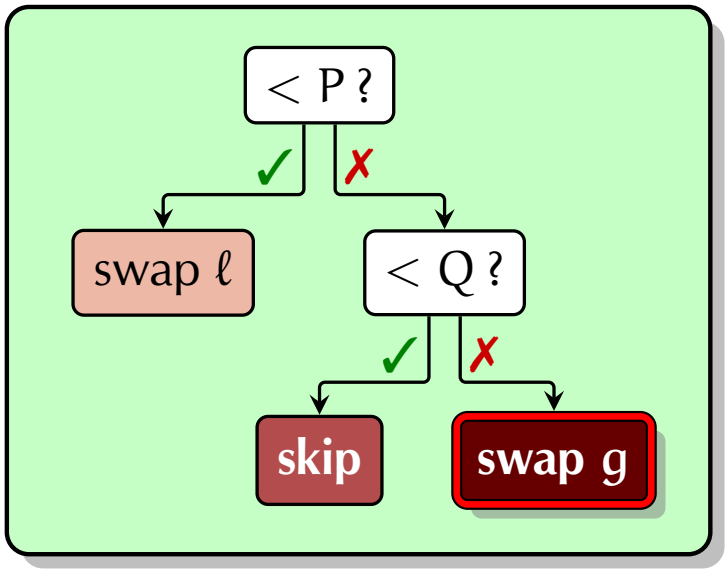


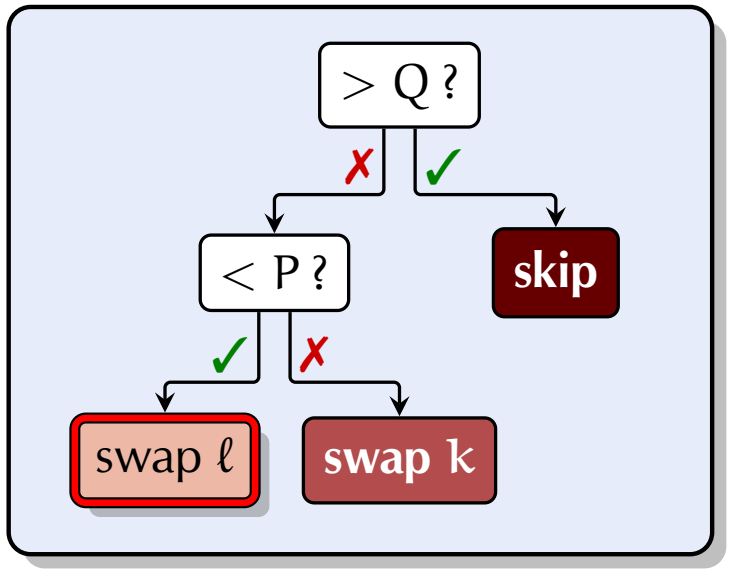


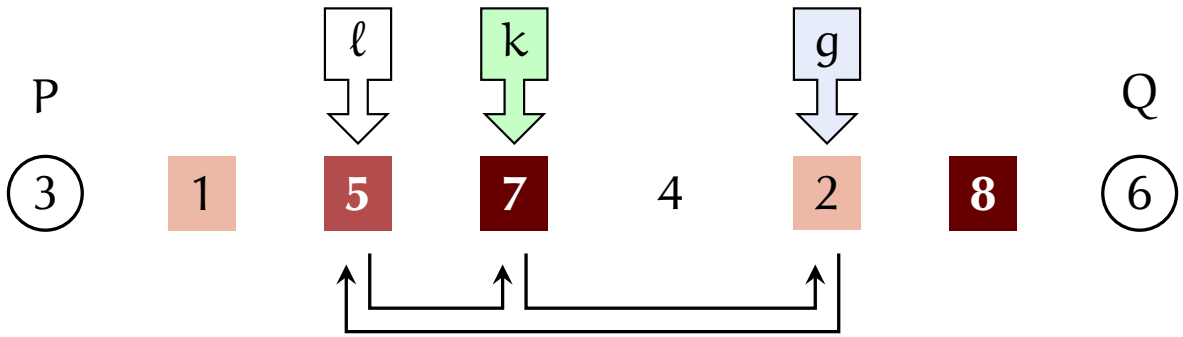
Invariant:



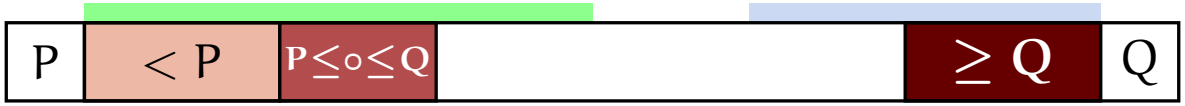
Yaroslavskiy's Algorithm



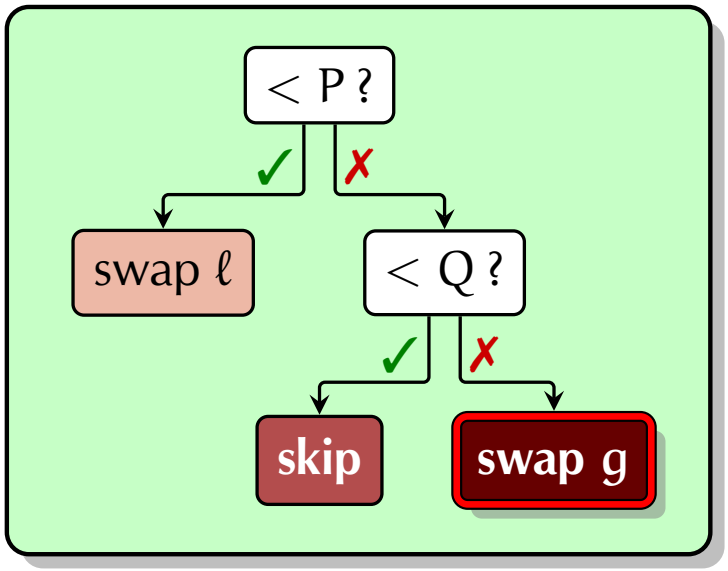


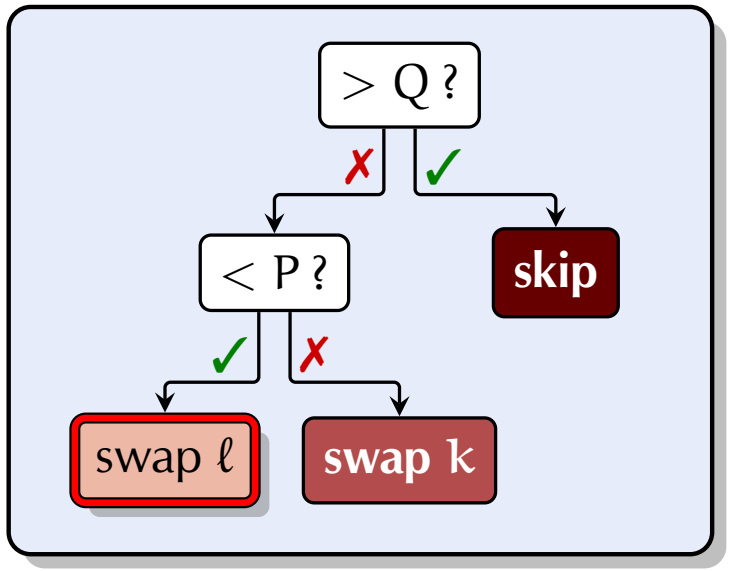


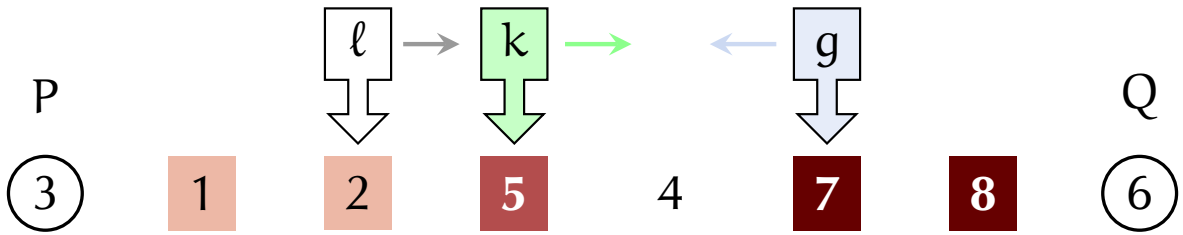
Invariant:



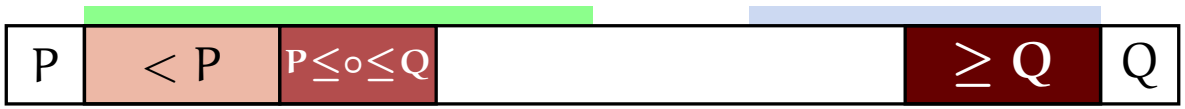
Yaroslavskiy's Algorithm



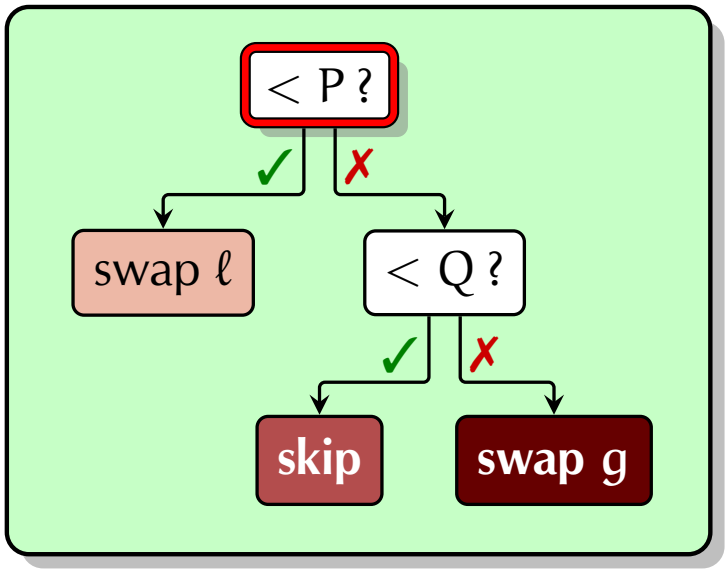


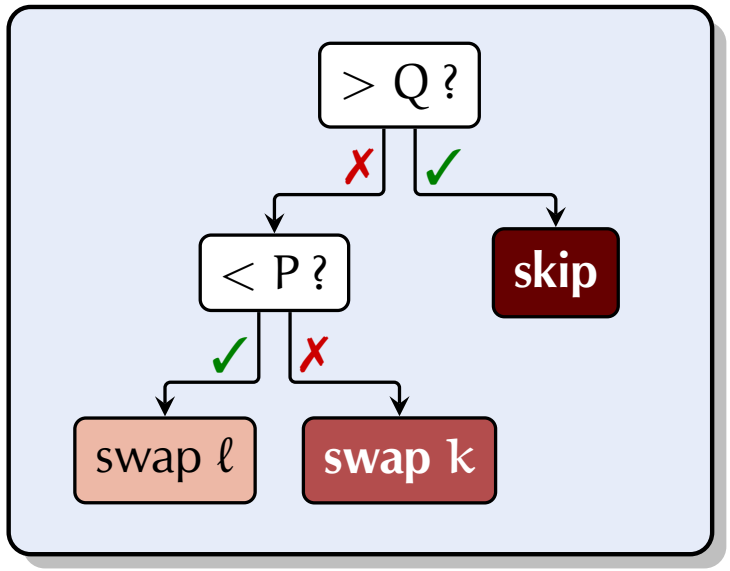


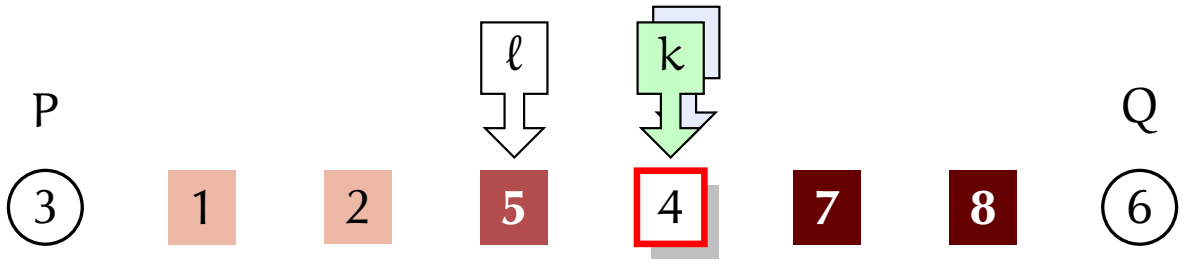
Invariant:



Yaroslavskiy's Algorithm



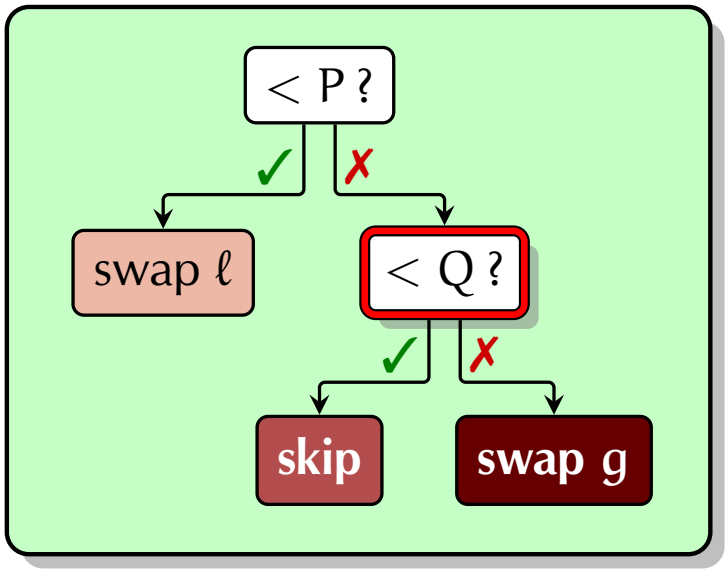


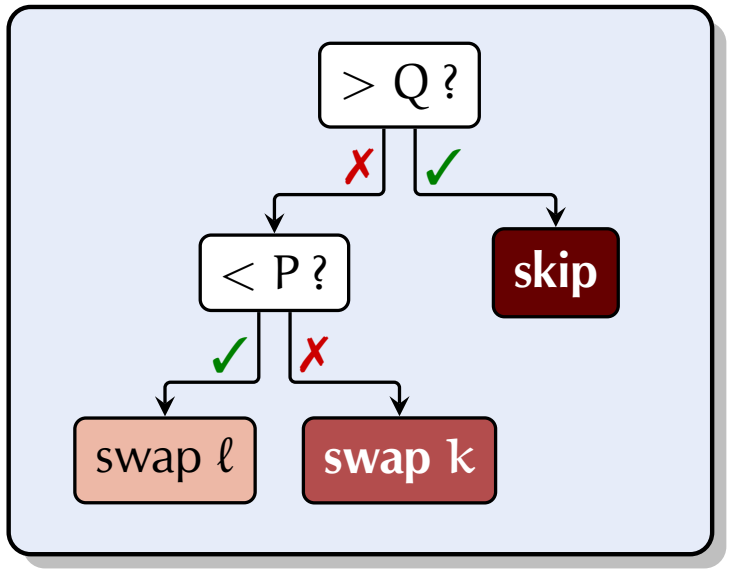


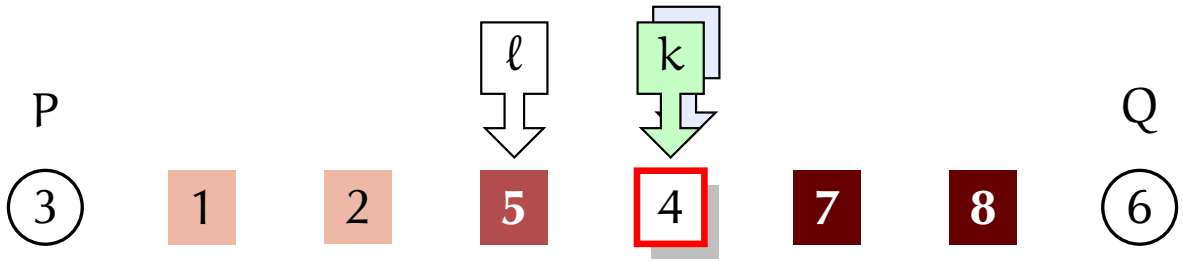
Invariant:



Yaroslavskiy's Algorithm



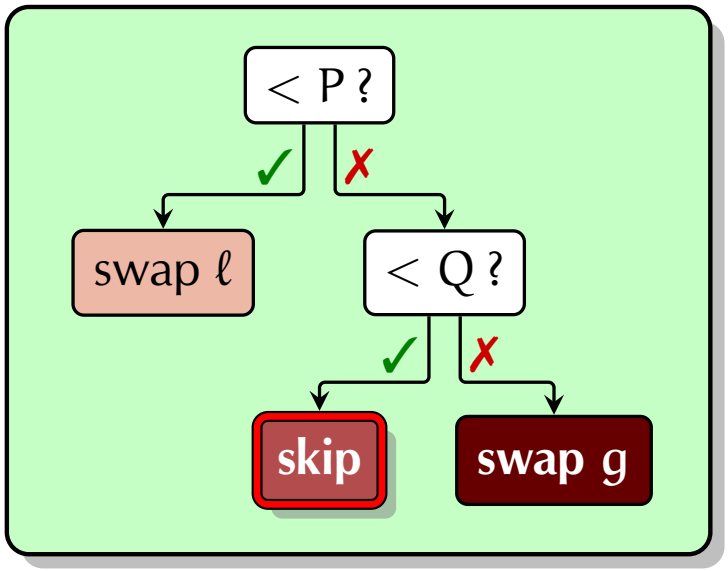


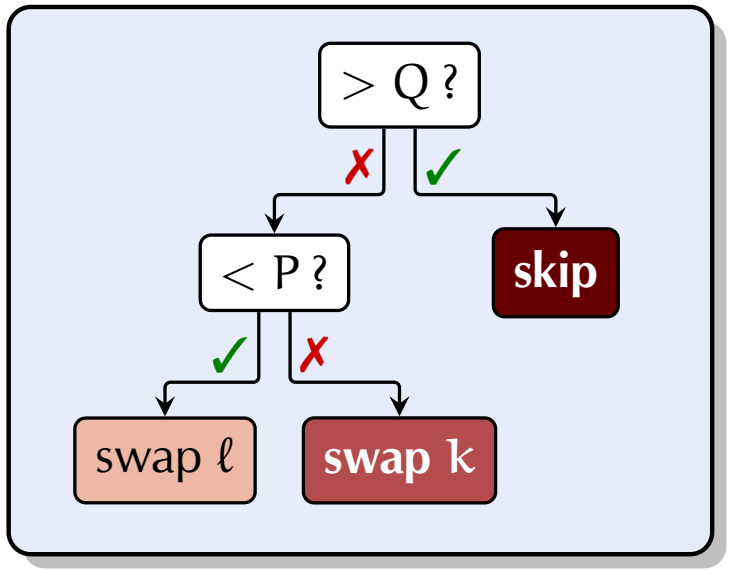


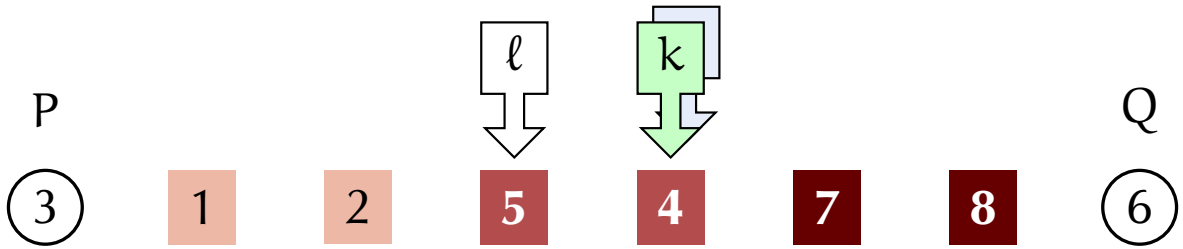
Invariant:



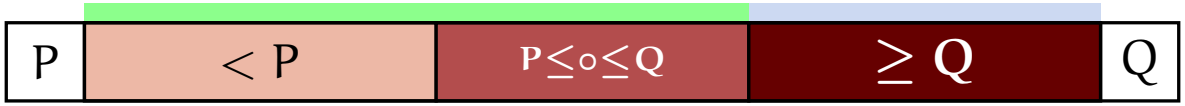
Yaroslavskiy's Algorithm



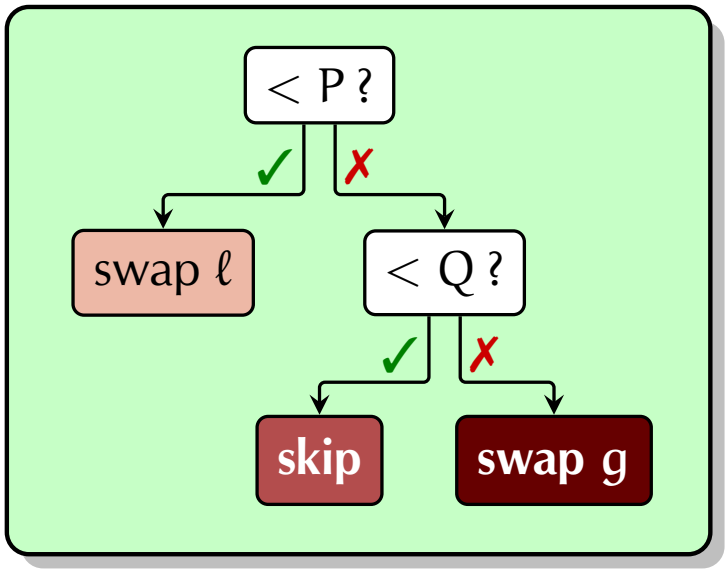


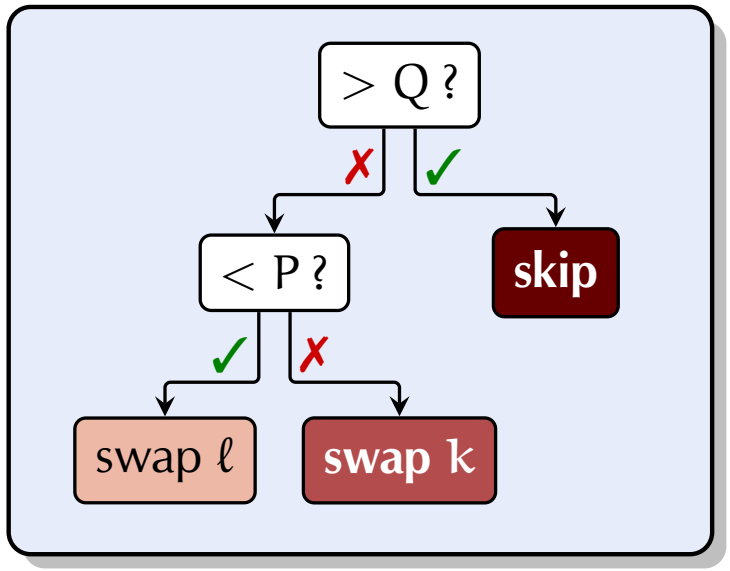


Invariant:



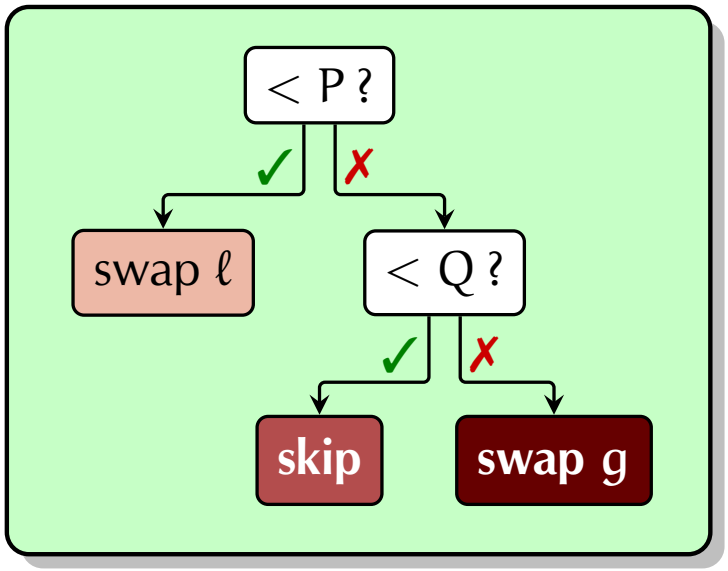
Yaroslavskiy's Algorithm

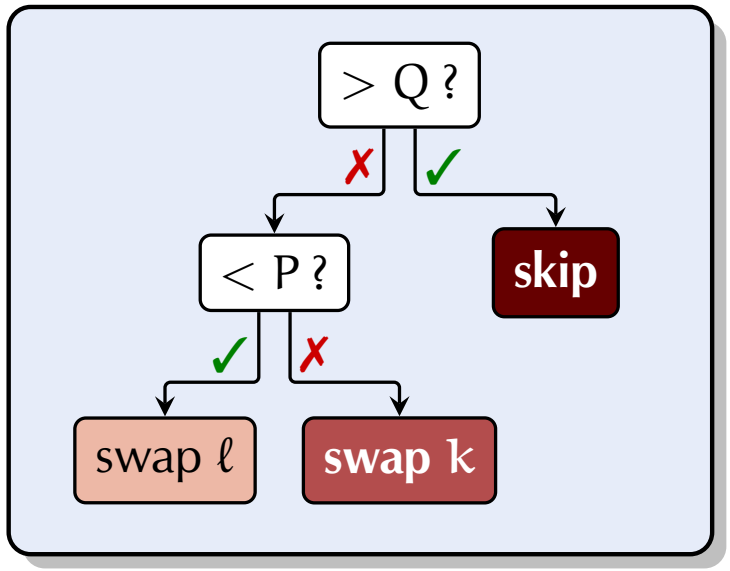


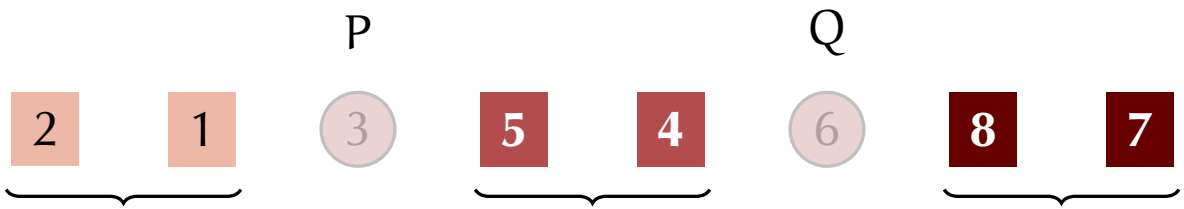




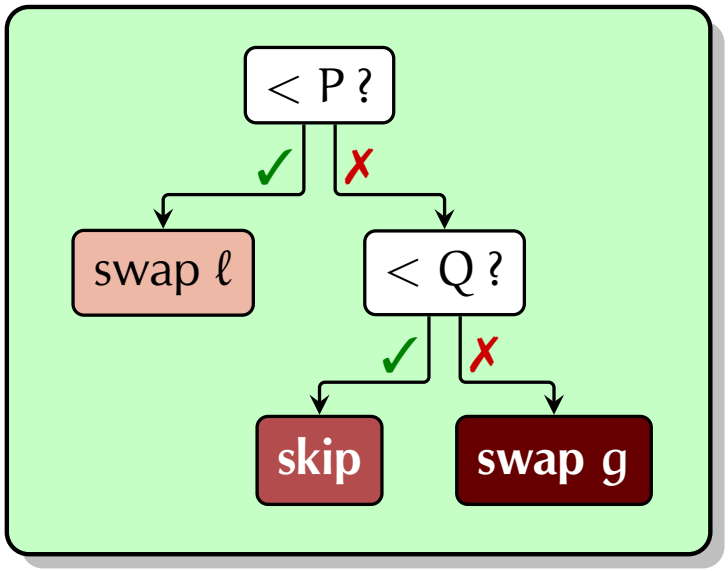
Yaroslavskiy's Algorithm

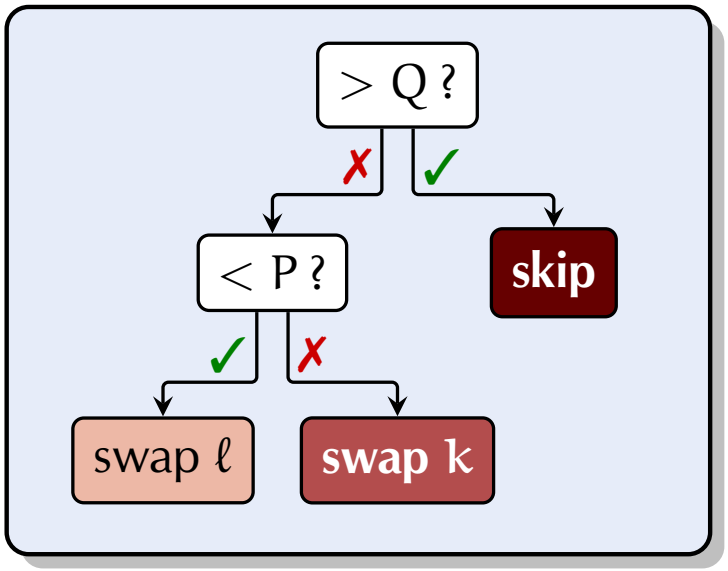






Yaroslavskiy's Algorithm







Pivot Sampling

- Classic Quicksort: pivot as **median of k**
  (in practice: $k = 3$, pseudomedian of 9)
- We need **two** pivots.
- **Here:** parametric scheme with parameter $t = (t_1, t_2, t_3)$
  1. Sample $k = t_1 + t_2 + t_3 + 2$ elements
  2. **Sort** the sample
  3. Select pivots s. t. in sorted sample

     t_1 smaller,
     t_2 between and
     t_3 larger than pivots

Example with
$k = 8$ and $t = (3, 2, 1)$:

- **ALENEX 2013**:
  Empirical evidence that **asymmetric** $t = (0, 1, 2)$ beats $t = (1, 1, 1)$!

Pivot Sampling

- Classic Quicksort: pivot as **median of k**
  (in practice: $k = 3$, pseudomedian of 9)
- We need **two** pivots.
- **Here:** parametric scheme with parameter $\mathbf{t} = (t_1, t_2, t_3)$
  1. Sample $k = t_1 + t_2 + t_3 + 2$ elements
  2. **Sort** the sample
  3. Select pivots s. t. in sorted sample

     t_1 smaller,
     t_2 between and
     t_3 larger than pivots

  Example with
  $k = 8$ and $\mathbf{t} = (3, 2, 1)$:

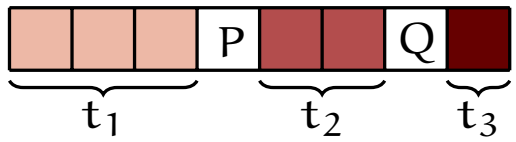
- **ALENEX 2013**:
  Empirical evidence that **asymmetric** $\mathbf{t} = (0, 1, 2)$ beats $\mathbf{t} = (1, 1, 1)$!

Pivot Sampling

- Classic Quicksort: pivot as **median of k**
  (in practice: $k = 3$, pseudomedian of 9)
- We need **two** pivots.
- **Here:** parametric scheme with parameter $\mathbf{t} = (t_1, t_2, t_3)$
  1. Sample $k = t_1 + t_2 + t_3 + 2$ elements
  2. **Sort** the sample
  3. Select pivots s. t. in sorted sample

t_1 smaller,
t_2 between and
t_3 larger than pivots

Example with
$k = 8$ and $\mathbf{t} = (3, 2, 1)$:



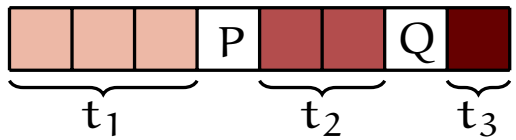
- **ALENEX 2013:**
  Empirical evidence that **asymmetric** $\mathbf{t} = (0, 1, 2)$ beats $\mathbf{t} = (1, 1, 1)$!

Pivot Sampling

- Classic Quicksort: pivot as **median of k**
  (in practice: $k = 3$, pseudomedian of 9)
- We need **two** pivots.
- **Here:** parametric scheme with parameter $\mathbf{t} = (t_1, t_2, t_3)$
  1. Sample $k = t_1 + t_2 + t_3 + 2$ elements
  2. **Sort** the sample
  3. Select pivots s. t. in sorted sample

t_1 smaller,
t_2 between and
t_3 larger than pivots

Example with
$k = 8$ and $\mathbf{t} = (3, 2, 1)$:



- **ALENEX 2013**:
  Empirical evidence that **asymmetric** $\mathbf{t} = (0, 1, 2)$ beats $\mathbf{t} = (1, 1, 1)$!

Expected Number of Comparisons

Goal: *Count* ***comparisons*** *in Dual-Pivot Quicksort with Pivot Sampling*

C_n: (random!) #cmps to sort n **i. i. d.** ***Uniform***$(0,1)$ elements

$\rightsquigarrow$ **Distributional Recurrence:**

$$C_n \overset{\mathcal{D}}{=} T_n + C_{J_1} + C_{J_2} + C_{J_3}$$

- $J = (J_1, J_2, J_3)$: (random) subproblem sizes
- T_n: (random) #cmps of first partitioning step

$\rightsquigarrow$ Taking expectations and conditioning on J:

$$\mathbb{E}[C_n] = \mathbb{E}[T_n] + \sum_{j=0}^{n-2} w_{n,j}\, \mathbb{E}[C_j]$$

$$\text{with} \quad w_{n,j} = \Pr[J_1 = j] + \Pr[J_2 = j] + \Pr[J_3 = j]$$

Expected Number of Comparisons

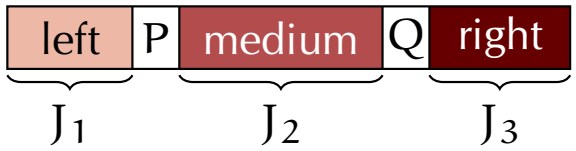
Goal: *Count* ***comparisons*** *in Dual-Pivot Quicksort with Pivot Sampling*

C_n: (random!) #cmps to sort n **i. i. d.** ***Uniform***$(0,1)$ elements

$\rightsquigarrow$ **Distributional Recurrence:**

$$C_n \overset{\mathcal{D}}{=} T_n + C_{J_1} + C_{J_2} + C_{J_3}$$

- $J = (J_1, J_2, J_3)$: (random) subproblem sizes
- T_n: (random) #cmps of first partitioning step



$\rightsquigarrow$ Taking expectations and conditioning on J:

$$\mathbb{E}[C_n] = \mathbb{E}[T_n] + \sum_{j=0}^{n-2} w_{n,j}\, \mathbb{E}[C_j]$$

$$\text{with} \quad w_{n,j} = \Pr[J_1 = j] + \Pr[J_2 = j] + \Pr[J_3 = j]$$

Expected Number of Comparisons

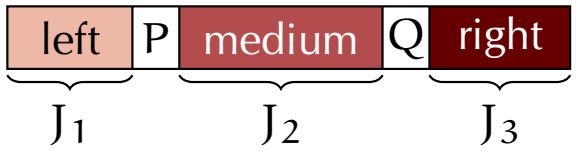
Goal: *Count* ***comparisons*** *in Dual-Pivot Quicksort with Pivot Sampling*

C_n: (random!) #cmps to sort n **i. i. d.** ***Uniform***$(0,1)$ elements

$\rightsquigarrow$ **Distributional Recurrence:**

$$C_n \overset{\mathcal{D}}{=} T_n + C_{J_1} + C_{J_2} + C_{J_3}$$

- $\mathbf{J} = (J_1, J_2, J_3)$: (random) subproblem sizes
- T_n: (random) #cmps of first partitioning step



$\rightsquigarrow$ Taking expectations and conditioning on $\mathbf{J}$:

$$\mathbb{E}[C_n] = \mathbb{E}[T_n] + \sum_{j=0}^{n-2} w_{n,j}\, \mathbb{E}[C_j]$$

$$\text{with} \quad w_{n,j} = \Pr[J_1 = j] + \Pr[J_2 = j] + \Pr[J_3 = j]$$

Solution of Recurrence

Recurrence: $\mathbb{E}[C_n] = \mathbb{E}[T_n] + \sum_{j=0}^{n-2} w_{n,j}\,\mathbb{E}[C_j]$ with $\mathbb{E}[T_n] \sim \gamma n$

Apply ***Continuous Master Theorem*** [Roura 2001]

Idea: Approximate sum by **integral**

$$\sum_{j=0}^{n-2} w_{n,j}\,\mathbb{E}[C_j] \approx \int_0^n w_{n,j}\,\mathbb{E}[C_j]\,dj \approx \int_0^1 n \cdot w_{n,\alpha n}\,\mathbb{E}[C_{\alpha n}]\,d\alpha$$

$\rightsquigarrow$ Continuous Recurrence $\mathbb{E}[C_n] \approx \mathbb{E}[T_n] + \int_0^1 w(\alpha)\,\mathbb{E}[C_{\alpha n}]\,d\alpha$

Ansatz: $\mathbb{E}[C_n] \approx \frac{\gamma}{\mu} n \ln n$

$\rightsquigarrow$ Fulfils recurrence for $\mu = \sum_{k=0}^{\infty} \frac{k+1}{k+2}\,(\mathbb{E}[C_{k+1}] - \mathbb{E}[C_{k+2}])$

$\rightsquigarrow$ $\mathbb{E}[C_n] \approx \frac{\gamma}{\mu} n \ln n$

Solution of Recurrence

Recurrence: $\mathbb{E}[C_n] = \mathbb{E}[T_n] + \sum_{j=0}^{n-2} w_{n,j}\,\mathbb{E}[C_j] \quad \text{with} \quad \mathbb{E}[T_n] \sim \gamma n$

Apply ***Continuous Master Theorem*** [Roura 2001]

Idea: Approximate sum by **integral** over **relative** subprobem size $\alpha = j/n$

$$\sum_{j=0}^{n-2} w_{n,j}\,\mathbb{E}[C_j] \;\approx\; \int_0^n w_{n,j}\,\mathbb{E}[C_j]\,dj \;\approx\; \int_0^1 n \cdot w_{n,\alpha n}\,\mathbb{E}[C_{\alpha n}]\,d\alpha$$

$\rightsquigarrow$ **Continuous** Recurrence: $\mathbb{E}[C_n] \approx \mathbb{E}[T_n] + \int_0^1 w(\alpha)\,\mathbb{E}[C_{\alpha n}]\,d\alpha$

Ansatz: $\mathbb{E}[C_n] = \frac{\gamma}{H} n \ln n$

$\rightsquigarrow$ fulfills recurrence for $H = \sum_{\ell=1}^{3} \frac{t_\ell + 1}{k+1}\left(\mathcal{H}_{k+1} - \mathcal{H}_{t_\ell+1}\right)$

$\rightsquigarrow$ $\mathbb{E}[C_n] \sim \frac{\gamma}{H} n \ln n$

Next: Variance

Solution of Recurrence

Recurrence: $\mathbb{E}[C_n] = \mathbb{E}[T_n] + \sum_{j=0}^{n-2} w_{n,j}\,\mathbb{E}[C_j]$ with $\mathbb{E}[T_n] \sim \gamma n$

Apply ***Continuous Master Theorem*** [Roura 2001]

Idea: Approximate sum by **integral** over **relative** subprobem size $\alpha = j/n$

$$\sum_{j=0}^{n-2} w_{n,j}\,\mathbb{E}[C_j] \approx \int_0^n w_{n,j}\,\mathbb{E}[C_j]\,dj \approx \int_0^1 n \cdot w_{n,\alpha n}\,\mathbb{E}[C_{\alpha n}]\,d\alpha$$

$\rightsquigarrow$ **Continuous** Recurrence: $\mathbb{E}[C_n] \approx \mathbb{E}[T_n] + \int_0^1 w(\alpha)\,\mathbb{E}[C_{\alpha n}]\,d\alpha$

Ansatz: $\mathbb{E}[C_n] = \frac{\gamma}{H} n \ln n$

$\rightsquigarrow$ fulfills recurrence for $H = \sum_{t=1}^{3} \frac{t_t + 1}{k+1}\left(\mathcal{H}_{k+1} - \mathcal{H}_{t_t+1}\right)$

$\rightsquigarrow$ $\mathbb{E}[C_n] \sim \frac{\gamma}{H} n \ln n$

Next: Variance

Solution of Recurrence

Recurrence: $\mathbb{E}[C_n] = \mathbb{E}[T_n] + \sum_{j=0}^{n-2} w_{n,j}\,\mathbb{E}[C_j]$ with $\mathbb{E}[T_n] \sim \gamma n$

Apply ***Continuous Master Theorem*** [Roura 2001]

Idea: Approximate sum by **integral** over **relative** subprobem size $\alpha = j/n$

$$\sum_{j=0}^{n-2} w_{n,j}\,\mathbb{E}[C_j] \approx \int_0^n w_{n,j}\,\mathbb{E}[C_j]\,dj \approx \int_0^1 \underbrace{n \cdot w_{n,\alpha n}}_{\to w(\alpha)}\,\mathbb{E}[C_{\alpha n}]\,d\alpha$$

$\rightsquigarrow$ **Continuous** Recurrence: $\mathbb{E}[C_n] \approx \mathbb{E}[T_n] + \int_0^1 w(\alpha)\,\mathbb{E}[C_{\alpha n}]\,d\alpha$

Ansatz: $\mathbb{E}[C_n] = \frac{\gamma}{H} n \ln n$

$\rightsquigarrow$ fulfills recurrence for $H = \sum_{\ell=1}^{3} \frac{t_\ell + 1}{k+1}\left(\mathcal{H}_{k+1} - \mathcal{H}_{t_\ell+1}\right)$

$\rightsquigarrow$ $\mathbb{E}[C_n] \sim \frac{\gamma}{H} n \ln n$

Next: Variance

Solution of Recurrence

Recurrence: $\mathbb{E}[C_n] = \mathbb{E}[T_n] + \sum_{j=0}^{n-2} w_{n,j}\,\mathbb{E}[C_j] \quad \text{with} \quad \mathbb{E}[T_n] \sim \gamma n$

Apply ***Continuous Master Theorem*** [Roura 2001]

Idea: Approximate sum by **integral** over **relative** subprobem size $\alpha = j/n$

$$\sum_{j=0}^{n-2} w_{n,j}\,\mathbb{E}[C_j] \;\approx\; \int_0^n w_{n,j}\,\mathbb{E}[C_j]\,dj \;\approx\; \int_0^1 \underbrace{n \cdot w_{n,\alpha n}}_{\to\, w(\alpha)}\,\mathbb{E}[C_{\alpha n}]\,d\alpha$$

$\rightsquigarrow$ **Continuous** Recurrence: $\mathbb{E}[C_n] \approx \mathbb{E}[T_n] + \int_0^1 w(\alpha)\,\mathbb{E}[C_{\alpha n}]\,d\alpha$

Ansatz: $\mathbb{E}[C_n] = \frac{\gamma}{H} n \ln n$

$\rightsquigarrow$ fulfills recurrence for $H = \sum_{\ell=1}^{3} \frac{t_\ell + 1}{k+1}\left(\mathcal{H}_{k+1} - \mathcal{H}_{t_\ell+1}\right)$

$\rightsquigarrow$ $\mathbb{E}[C_n] \sim \frac{\gamma}{H} n \ln n$

Next: Analysis

Solution of Recurrence

Recurrence: $\mathbb{E}[C_n] = \mathbb{E}[T_n] + \sum_{j=0}^{n-2} w_{n,j}\,\mathbb{E}[C_j]$ with $\mathbb{E}[T_n] \sim \gamma n$

Apply ***Continuous Master Theorem*** [Roura 2001]

Idea: Approximate sum by **integral** over **relative** subprobem size $\alpha = j/n$

$$\sum_{j=0}^{n-2} w_{n,j}\,\mathbb{E}[C_j] \approx \int_0^n w_{n,j}\,\mathbb{E}[C_j]\,dj \approx \int_0^1 \underbrace{n \cdot w_{n,\alpha n}}_{\to w(\alpha)}\,\mathbb{E}[C_{\alpha n}]\,d\alpha$$

$\rightsquigarrow$ **Continuous** Recurrence: $\mathbb{E}[C_n] \approx \mathbb{E}[T_n] + \int_0^1 w(\alpha)\,\mathbb{E}[C_{\alpha n}]\,d\alpha$

Ansatz: $\mathbb{E}[C_n] = \frac{\gamma}{H} n \ln n$

$\rightsquigarrow$ fulfills recurrence for $H = \sum_{l=1}^{3} \frac{t_l+1}{k+1}\left(\mathcal{H}_{k+1} - \mathcal{H}_{t_l+1}\right)$

$\rightsquigarrow$ $\mathbb{E}[C_n] \sim \frac{\gamma}{H} n \ln n$

Solution of Recurrence

Recurrence: $\mathbb{E}[C_n] = \mathbb{E}[T_n] + \sum_{j=0}^{n-2} w_{n,j}\,\mathbb{E}[C_j]$ with $\mathbb{E}[T_n] \sim \gamma n$

Apply ***Continuous Master Theorem*** [Roura 2001]

Idea: Approximate sum by **integral** over **relative** subprobem size $\alpha = j/n$

$$\sum_{j=0}^{n-2} w_{n,j}\,\mathbb{E}[C_j] \approx \int_0^n w_{n,j}\,\mathbb{E}[C_j]\,dj \approx \int_0^1 \underbrace{n \cdot w_{n,\alpha n}}_{\to w(\alpha)}\,\mathbb{E}[C_{\alpha n}]\,d\alpha$$

$\rightsquigarrow$ **Continuous** Recurrence: $\mathbb{E}[C_n] \approx \mathbb{E}[T_n] + \int_0^1 w(\alpha)\,\mathbb{E}[C_{\alpha n}]\,d\alpha$

Ansatz: $\mathbb{E}[C_n] = \frac{\gamma}{H} n \ln n$

$\rightsquigarrow$ fulfills recurrence for $H = \sum_{l=1}^{3} \frac{t_l + 1}{k+1}\left(\mathcal{H}_{k+1} - \mathcal{H}_{t_l+1}\right)$

$\rightsquigarrow$ $\mathbb{E}[C_n] \sim \frac{\gamma}{H} n \ln n$ *Next: What is γ?*

Solution of Recurrence

Recurrence: $\mathbb{E}[C_n] = \mathbb{E}[T_n] + \sum_{j=0}^{n-2} w_{n,j}\,\mathbb{E}[C_j]$ with $\mathbb{E}[T_n] \sim \gamma n$

Apply ***Continuous Master Theorem*** [Roura 2001]

Idea: Approximate sum by **integral** over **relative** subprobem size $\alpha = j/n$

$$\sum_{j=0}^{n-2} w_{n,j}\,\mathbb{E}[C_j] \approx \int_0^n w_{n,j}\,\mathbb{E}[C_j]\,dj \approx \int_0^1 \underbrace{n \cdot w_{n,\alpha n}}_{\to w(\alpha)}\,\mathbb{E}[C_{\alpha n}]\,d\alpha$$

$\rightsquigarrow$ **Continuous** Recurrence: $\mathbb{E}[C_n] \approx \mathbb{E}[T_n] + \int_0^1 w(\alpha)\,\mathbb{E}[C_{\alpha n}]\,d\alpha$

Ansatz: $\mathbb{E}[C_n] = \frac{\gamma}{H} n \ln n$

$\rightsquigarrow$ fulfills recurrence for $H = \sum_{l=1}^{3} \frac{t_l + 1}{k+1}\left(\mathcal{H}_{k+1} - \mathcal{H}_{t_l+1}\right)$

$\rightsquigarrow$ $\boxed{\mathbb{E}[C_n] \sim \frac{\gamma}{H} n \ln n}$

Next: What is γ?

Solution of Recurrence

Recurrence: $\mathbb{E}[C_n] = \mathbb{E}[T_n] + \sum_{j=0}^{n-2} w_{n,j}\,\mathbb{E}[C_j]$ with $\mathbb{E}[T_n] \sim \gamma n$

Apply ***Continuous Master Theorem*** [Roura 2001]

Idea: Approximate sum by **integral** over **relative** subprobem size $\alpha = j/n$

$$\sum_{j=0}^{n-2} w_{n,j}\,\mathbb{E}[C_j] \approx \int_0^n w_{n,j}\,\mathbb{E}[C_j]\,dj \approx \int_0^1 \underbrace{n \cdot w_{n,\alpha n}}_{\to w(\alpha)}\,\mathbb{E}[C_{\alpha n}]\,d\alpha$$

$\rightsquigarrow$ **Continuous** Recurrence: $\mathbb{E}[C_n] \approx \mathbb{E}[T_n] + \int_0^1 w(\alpha)\,\mathbb{E}[C_{\alpha n}]\,d\alpha$

Ansatz: $\mathbb{E}[C_n] = \frac{\gamma}{H} n \ln n$

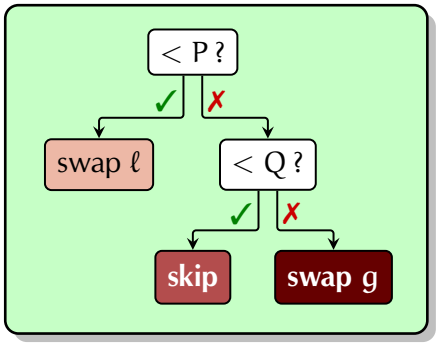
$\rightsquigarrow$ fulfills recurrence for $H = \sum_{l=1}^{3} \frac{t_l + 1}{k+1}\left(\mathcal{H}_{k+1} - \mathcal{H}_{t_l+1}\right)$

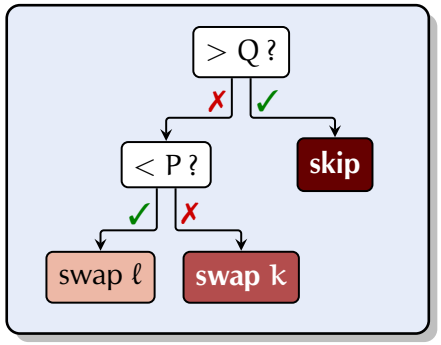
$\rightsquigarrow$ $\boxed{\mathbb{E}[C_n] \sim \frac{\gamma}{H} n \ln n}$

Next: *What is γ?*

#Comparisons per Partitioning

*How many comparisons for **one element**?* *It depends! Either 2 or 1.*





$\rightsquigarrow T_n \sim 2n$ – "cheap elements":

- s @ $\mathcal{K}$: small in k's range
- l @ $\mathcal{G}$: large in g's range

$\rightsquigarrow$ *need ranges* $\mathcal{K}$ *and* $\mathcal{G}$*!*

Recall **Invariant:**

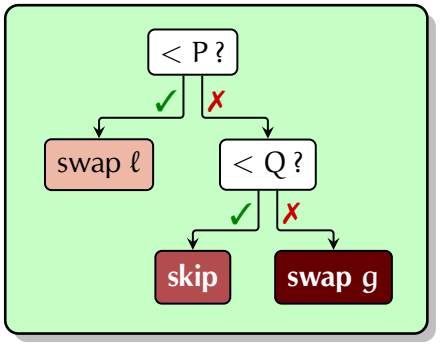
P	$< P$	$P \le \circ \le Q$	$\ge Q$	Q

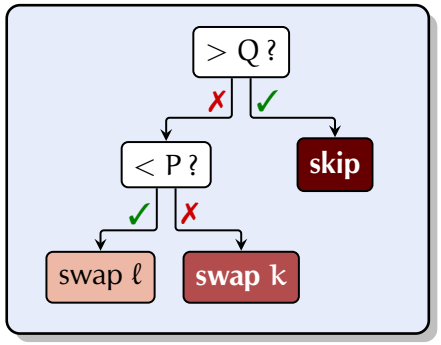
- $|\mathcal{K}| \sim I_1 + I_2$
- $|\mathcal{G}| \sim I_3$

with $I = (I_1, I_2, I_3)$ sizes of partitions

#Comparisons per Partitioning

How many comparisons for ***one element****?* *It depends! Either* 2 *or* 1 .





$\rightsquigarrow T_n \sim 2n$ – "cheap elements":

- s @ $\mathcal{K}$: small in k's range
- l @ $\mathcal{G}$: large in g's range

$\rightsquigarrow$ *need ranges* $\mathcal{K}$ *and* $\mathcal{G}$ *!*

Recall **Invariant:**

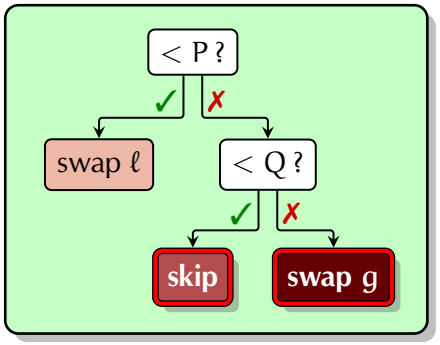
P	$< P$	$P \le \circ \le Q$	$\ge Q$	Q

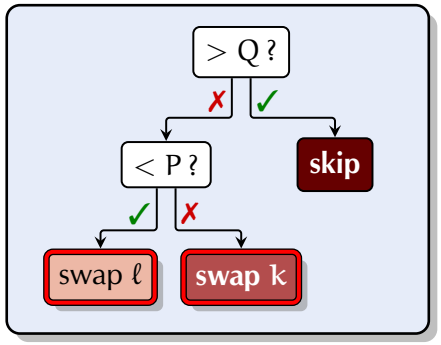
- $|\mathcal{K}| \sim I_1 + I_2$
- $|\mathcal{G}| \sim I_3$

with $I = (I_1, I_2, I_3)$ sizes of partitions

#Comparisons per Partitioning

*How many comparisons for **one element**? It depends! Either* 2 *or* 1 .





$\rightsquigarrow T_n \sim 2n$ – "cheap elements":

- s @ $\mathcal{K}$: small in k's range
- l @ $\mathcal{G}$: large in g's range

$\rightsquigarrow$ *need ranges* $\mathcal{K}$ *and* $\mathcal{G}$ *!*

Recall **Invariant:**

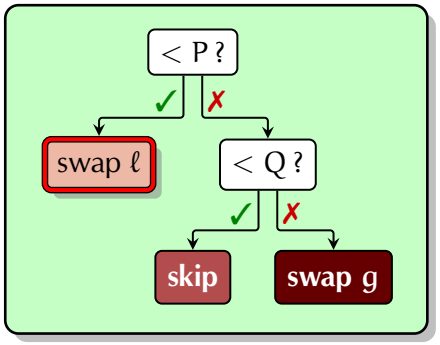
P	$< P$	$P \le \circ \le Q$	$\ge Q$	Q

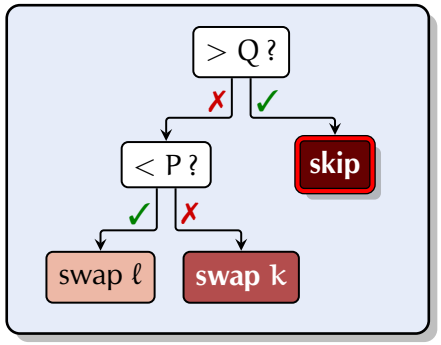
- $|\mathcal{K}| \sim I_1 + I_2$
- $|\mathcal{G}| \sim I_3$

with $I = (I_1, I_2, I_3)$ sizes of partitions

#Comparisons per Partitioning

*How many comparisons for **one element**? It depends! Either 2 or 1.*





$\rightsquigarrow T_n \sim 2n$ – "cheap elements":

- s @ $\mathcal{K}$: small in k's range
- l @ $\mathcal{G}$: large in g's range

$\rightsquigarrow$ *need ranges* $\mathcal{K}$ *and* $\mathcal{G}$ *!*

Recall **Invariant:**

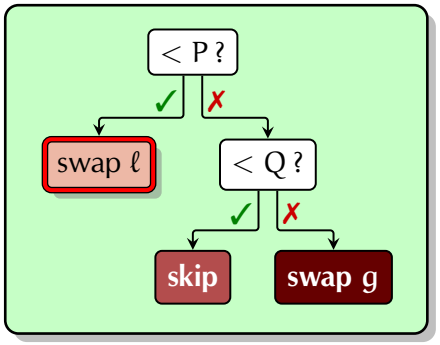
P	$< P$	$P \le \circ \le Q$	$\ge Q$	Q

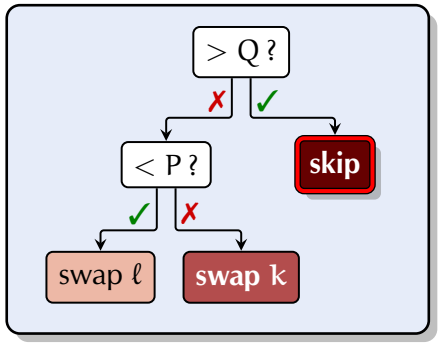
- $|\mathcal{K}| \sim I_1 + I_2$
- $|\mathcal{G}| \sim I_3$

with $I = (I_1, I_2, I_3)$ sizes of partitions

#Comparisons per Partitioning

*How many comparisons for **one element**? It depends! Either 2 or 1.*





$\rightsquigarrow T_n \sim 2n$ – "cheap elements":

- s @ $\mathcal{K}$: small in k's range
- ℓ @ $\mathcal{G}$: large in g's range

$\rightsquigarrow$ *need ranges* $\mathcal{K}$ *and* $\mathcal{G}$ *!*

Recall **Invariant:**

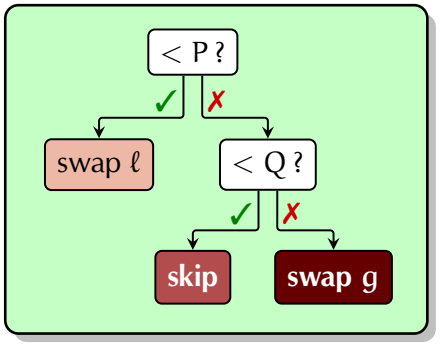
P	$< P$	$P \le \circ \le Q$	$\ge Q$	Q

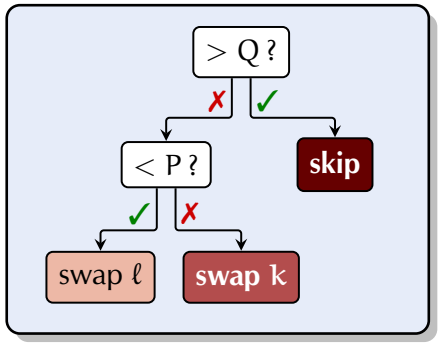
- $|\mathcal{K}| \sim I_1 + I_2$
- $|\mathcal{G}| \sim I_3$

with $I = (I_1, I_2, I_3)$ sizes of partitions

#Comparisons per Partitioning

*How many comparisons for **one element**? It depends! Either 2 or 1.*



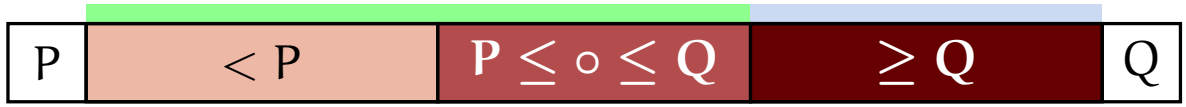


$\rightsquigarrow T_n \sim 2n$ – "cheap elements":

- s @ $\mathcal{K}$: small in k's range
- ℓ @ $\mathcal{G}$: large in g's range

$\rightsquigarrow$ *need ranges* $\mathcal{K}$ *and* $\mathcal{G}$ *!*

Recall **Invariant:**

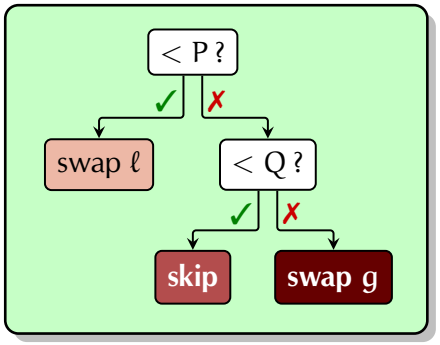


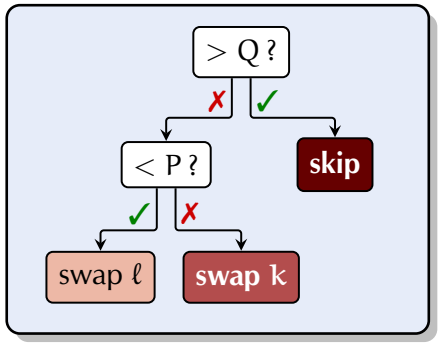
- $|\mathcal{K}| \sim I_1 + I_2$
- $|\mathcal{G}| \sim I_3$

with $I = (I_1, I_2, I_3)$ sizes of partitions

#Comparisons per Partitioning

*How many comparisons for **one element**? It depends! Either 2 or 1.*



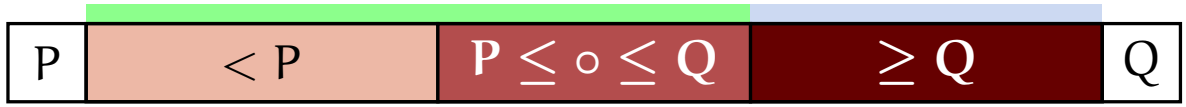


$\rightsquigarrow T_n \sim 2n$ – "cheap elements":

- s @ $\mathcal{K}$: small in k's range
- ℓ @ $\mathcal{G}$: large in g's range

$\rightsquigarrow$ *need ranges* $\mathcal{K}$ *and* $\mathcal{G}$ *!*

Recall **Invariant:**



- $|\mathcal{K}| \sim I_1 + I_2$
- $|\mathcal{G}| \sim I_3$

with $\mathbf{I} = (I_1, I_2, I_3)$ sizes of partitions

Distribution of s@$\mathcal{K}$ and l@$\mathcal{G}$

Consider $\mathbf{I}$ ***fixed***.

- $|\mathcal{K}| \sim I_1 + I_2$
- $|\mathcal{G}| \sim I_3$
- #small $= I_1$
- #large $= I_3$

Draw **positions** of small elements:

1. All array **indices** in urn
2. Draw without replacement
3. Count green balls

$\mathbf{I} = (3, 2, 3)$

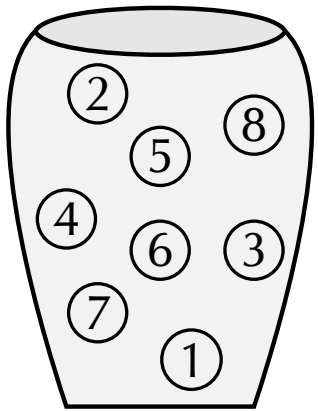
Distribution of s@$\mathcal{K}$ and l@$\mathcal{G}$

Consider $\mathbf{I}$ ***fixed***.

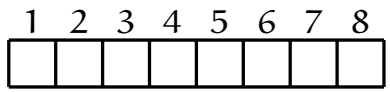
- $|\mathcal{K}| \sim I_1 + I_2$
- $|\mathcal{G}| \sim I_3$
- #small $= I_1$
- #large $= I_3$

Draw **positions** of small elements:

1. All array **indices** in urn
2. Draw without replacement
3. Count **green** balls



$\mathbf{I} = (3, 2, 3)$



$\rightsquigarrow$ s@$\mathcal{K}$ $= 2$

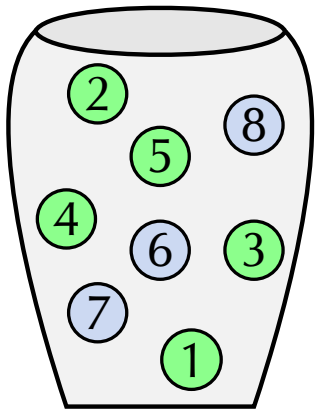
Distribution of s@$\mathcal{K}$ and l@$\mathcal{G}$

Consider $\mathbf{I}$ ***fixed***.

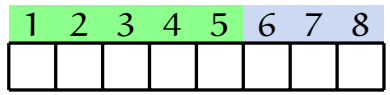
- $|\mathcal{K}| \sim I_1 + I_2$
- $|\mathcal{G}| \sim I_3$
- #small $= I_1$
- #large $= I_3$

Draw **positions** of small elements:

1. All array **indices** in urn
2. Draw without replacement
3. Count **green** balls



$\mathbf{I} = (3, 2, 3)$



$\rightsquigarrow$ s@$\mathcal{K}$ $= 2$

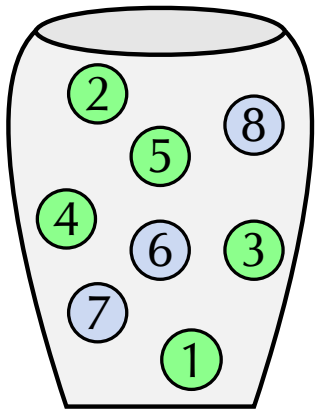
Distribution of $s@\mathcal{K}$ and $l@\mathcal{G}$

Consider $\mathbf{I}$ ***fixed***.

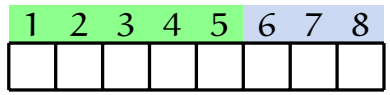
- $|\mathcal{K}| \sim I_1 + I_2$
- $|\mathcal{G}| \sim I_3$
- #small $= I_1$
- #large $= I_3$

Draw **positions** of small elements:

1. All array **indices** in urn
2. Draw without replacement
3. Count **green** balls



$\mathbf{I} = (3, 2, 3)$



$\rightsquigarrow s@\mathcal{K} = 2$

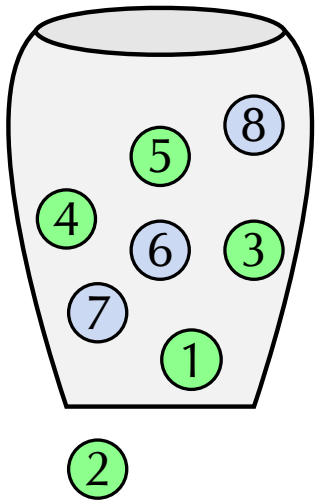
Distribution of s@$\mathcal{K}$ and l@$\mathcal{G}$

Consider $\mathbf{I}$ ***fixed***.

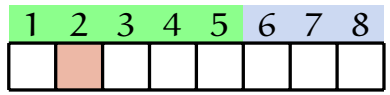
- $|\mathcal{K}| \sim I_1 + I_2$
- $|\mathcal{G}| \sim I_3$
- #small $= I_1$
- #large $= I_3$

Draw **positions** of small elements:

1. All array **indices** in urn
2. Draw without replacement
3. Count **green** balls



$\mathbf{I} = (3, 2, 3)$



$\rightsquigarrow$ s@$\mathcal{K}$ $= 2$

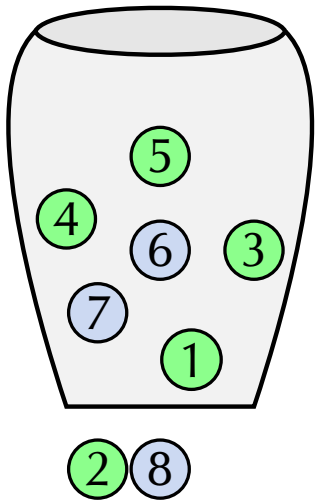
Distribution of $s@\mathcal{K}$ and $l@\mathcal{G}$

Consider $\mathbf{I}$ ***fixed***.

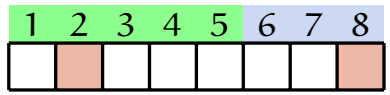
- $|\mathcal{K}| \sim I_1 + I_2$
- $|\mathcal{G}| \sim I_3$
- #small $= I_1$
- #large $= I_3$

Draw **positions** of small elements:

1. All array **indices** in urn
2. Draw without replacement
3. Count **green** balls



$\mathbf{I} = (3, 2, 3)$



$\rightsquigarrow s@\mathcal{K} = 2$

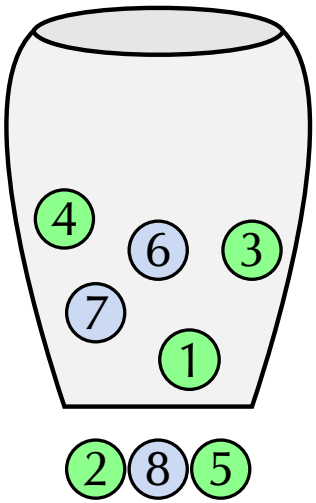
Distribution of s@$\mathcal{K}$ and l@$\mathcal{G}$

Consider $\mathbf{I}$ ***fixed***.

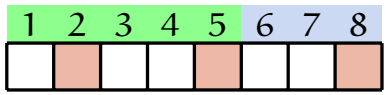
- $|\mathcal{K}| \sim I_1 + I_2$
- $|\mathcal{G}| \sim I_3$
- #small $= I_1$
- #large $= I_3$

Draw **positions** of small elements:

1. All array **indices** in urn
2. Draw without replacement
3. Count **green** balls



$\mathbf{I} = (3, 2, 3)$



$\rightsquigarrow$ s@$\mathcal{K}$ $= 2$

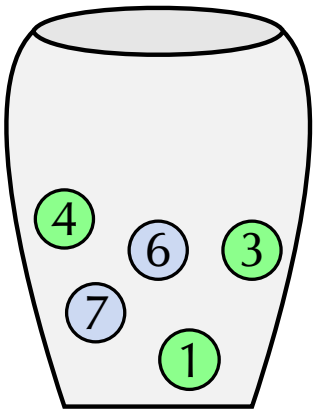
Distribution of s@$\mathcal{K}$ and l@$\mathcal{G}$

Consider $\mathbf{I}$ ***fixed***.

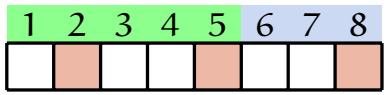
- $|\mathcal{K}| \sim I_1 + I_2$
- $|\mathcal{G}| \sim I_3$
- #small $= I_1$
- #large $= I_3$

Draw **positions** of small elements:

1. All array **indices** in urn
2. Draw without replacement
3. Count **green** balls



$\mathbf{I} = (3, 2, 3)$



2 8 5 $\rightsquigarrow$ s@$\mathcal{K}$ $= 2$

Distribution of $s@\mathcal{K}$ and $l@\mathcal{G}$

Consider $\mathbf{I}$ ***fixed***.

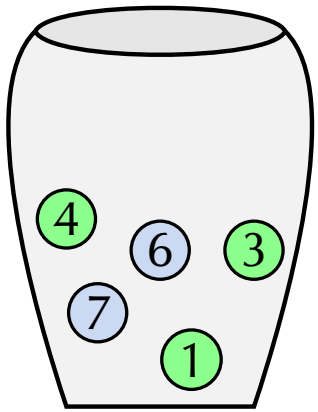
- $|\mathcal{K}| \sim I_1 + I_2$
- $|\mathcal{G}| \sim I_3$
- #small $= I_1$
- #large $= I_3$

$$s@\mathcal{K} \overset{\mathcal{D}}{=} \textit{Hypergeometric}(\#\text{small}, |\mathcal{K}|, n-k)$$

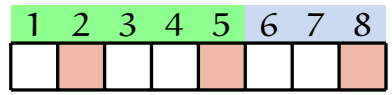
[sample excluded from partitioning $\rightsquigarrow n - k$]

Draw **positions** of small elements:

1. All array **indices** in urn
2. Draw without replacement
3. Count **green** balls



$$\mathbf{I} = (3, 2, 3)$$



2 8 5 $\rightsquigarrow s@\mathcal{K} = 2$

Similarly: $l@\mathcal{G} \overset{\mathcal{D}}{=} \textit{Hypergeometric}(\#\text{large}, |\mathcal{G}|, n-k)$

Distribution of s@$\mathcal{K}$ and l@$\mathcal{G}$

Consider $\mathbf{I}$ ***fixed***.

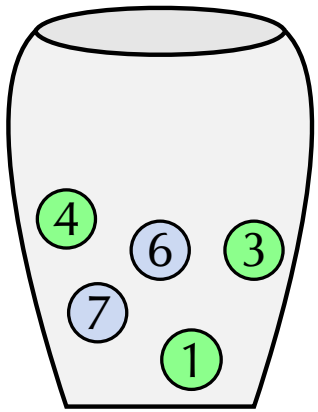
- $|\mathcal{K}| \sim I_1 + I_2$
- $|\mathcal{G}| \sim I_3$
- #small $= I_1$
- #large $= I_3$

$$s@\mathcal{K} \overset{\mathcal{D}}{=} \textit{Hypergeometric}(\#\text{small}, |\mathcal{K}|, n-k)$$

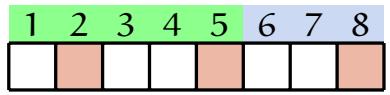
[sample excluded from partitioning $\rightsquigarrow n - k$]

Draw **positions** of small elements:

1. All array **indices** in urn
2. Draw without replacement
3. Count **green** balls



$\mathbf{I} = (3, 2, 3)$



2 8 5 $\rightsquigarrow s@\mathcal{K} = 2$

Similarly: l@$\mathcal{G}$ $\overset{\mathcal{D}}{=}$ *Hypergeometric*(#large, $|\mathcal{G}|$, $n-k$)

Distribution of Partition Sizes I

Recall: *Input* n ***i. i. d.*** *Uniform*$(0,1)$ *variables.*

- Assume **fixed** **pivot values** P and Q

$\rightsquigarrow$ **Classes** of elements are **i. i. d.** with

$$\Pr[x \text{ small }] = P$$
$$\Pr[x \text{ medium }] = Q - P$$
$$\Pr[x \text{ large }] = 1 - Q$$

- **Nicer:** In terms of **Spacings** $D = (D_1, D_2, D_3)$



$\rightsquigarrow$ Partition sizes I = cumulative result of independent repetitions

Distribution of Partition Sizes I

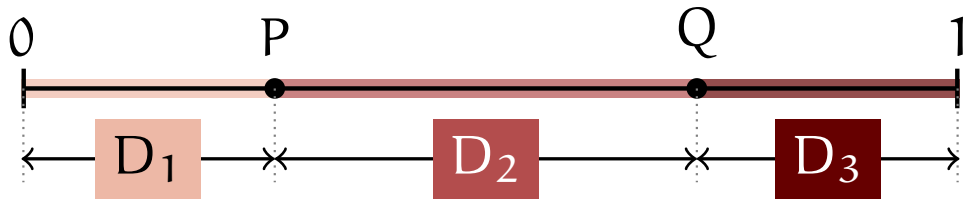
Recall: *Input* π ***i. i. d.*** *Uniform*$(0,1)$ *variables.*

- Assume **fixed** **pivot values** P and Q

$\rightsquigarrow$ **Classes** of elements are **i. i. d.** with

$$\Pr[x \text{ small }] = P \quad =: D_1$$
$$\Pr[x \text{ medium }] = Q - P \quad =: D_2$$
$$\Pr[x \text{ large }] = 1 - Q \quad =: D_3$$

- **Nicer:** In terms of **Spacings** $D = (D_1, D_2, D_3)$



$\rightsquigarrow$ Partition sizes I = cumulative result of independent repetitions

Distribution of Partition Sizes I

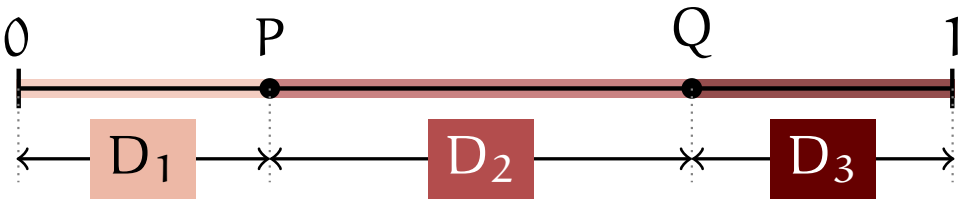
Recall: *Input* n ***i. i. d.*** *Uniform*$(0,1)$ *variables.*

- Assume **fixed** **pivot values** P and Q

$\rightsquigarrow$ **Classes** of elements are **i. i. d.** with

$$\Pr[x \text{ small }] = P \quad =: D_1$$
$$\Pr[x \text{ medium }] = Q - P \quad =: D_2$$
$$\Pr[x \text{ large }] = 1 - Q \quad =: D_3$$

- **Nicer:** In terms of **Spacings** $\mathbf{D} = (D_1, D_2, D_3)$



$\rightsquigarrow$ Partition sizes $\mathbf{I}$ = cumulative result of independent repetitions

$$\mathbf{I} \overset{\mathcal{D}}{=} \mathit{Multinomial}(n-k, \mathbf{D})$$

Distribution of Partition Sizes I

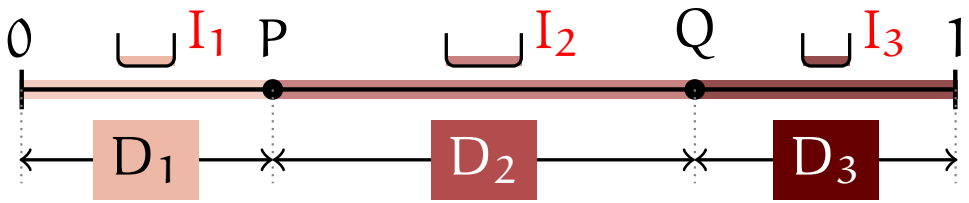
Recall: *Input* n ***i. i. d.*** *Uniform*$(0,1)$ *variables.*

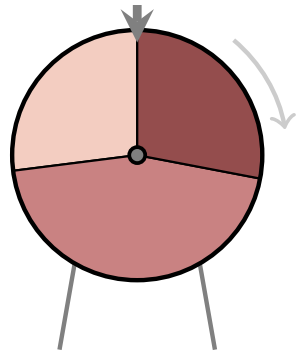
- Assume **fixed** **pivot values** P and Q

$\rightsquigarrow$ **Classes** of elements are **i. i. d.** with

$$\Pr[x \text{ small }] = P \quad =: D_1$$
$$\Pr[x \text{ medium }] = Q - P \quad =: D_2$$
$$\Pr[x \text{ large }] = 1 - Q \quad =: D_3$$

- **Nicer:** In terms of **Spacings** $\mathbf{D} = (D_1, D_2, D_3)$





$\rightsquigarrow$ Partition sizes $\mathbf{I}$ = cumulative result of independent repetitions

$$\mathbf{I} \overset{\mathcal{D}}{=} \mathit{Multinomial}(n-k, \mathbf{D})$$

Distribution of Partition Sizes I

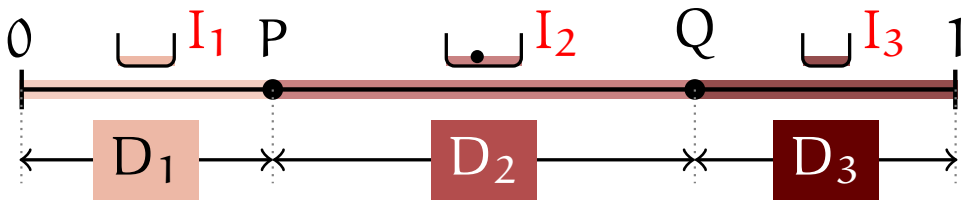
Recall: *Input* n ***i. i. d.*** *Uniform*$(0,1)$ *variables.*

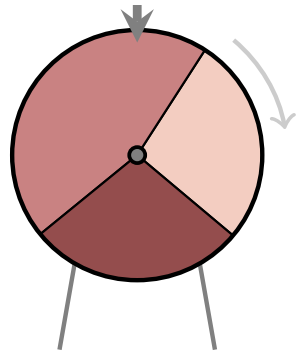
- Assume **fixed** **pivot values** P and Q

$\rightsquigarrow$ **Classes** of elements are **i. i. d.** with

$$\Pr[x \text{ small }] = P \quad =: D_1$$
$$\Pr[x \text{ medium }] = Q - P \quad =: D_2$$
$$\Pr[x \text{ large }] = 1 - Q \quad =: D_3$$

- **Nicer:** In terms of **Spacings** $\mathbf{D} = (D_1, D_2, D_3)$





$\rightsquigarrow$ Partition sizes $\mathbf{I}$ = cumulative result of independent repetitions

$$\mathbf{I} \overset{\mathcal{D}}{=} \mathit{Multinomial}(n-k, \mathbf{D})$$

Distribution of Partition Sizes I

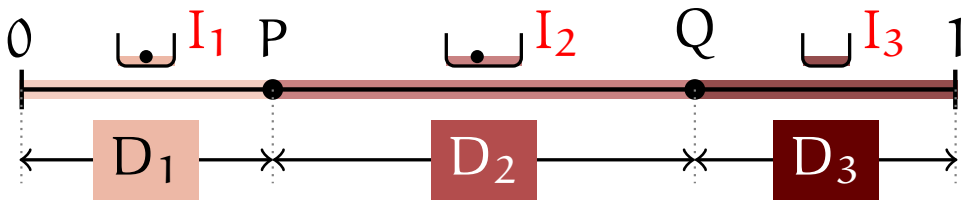
Recall: *Input* n ***i. i. d.*** *Uniform*$(0,1)$ *variables.*

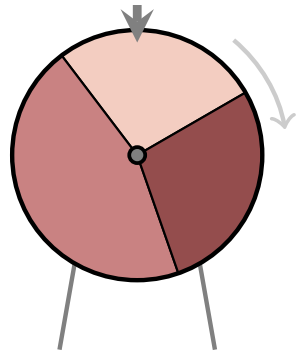
- Assume **fixed** **pivot values** P and Q

$\rightsquigarrow$ **Classes** of elements are **i. i. d.** with

$$\Pr[x \text{ small }] = P =: D_1$$
$$\Pr[x \text{ medium }] = Q - P =: D_2$$
$$\Pr[x \text{ large }] = 1 - Q =: D_3$$

- **Nicer:** In terms of **Spacings** $\mathbf{D} = (D_1, D_2, D_3)$





$\rightsquigarrow$ Partition sizes $\mathbf{I}$ = cumulative result of independent repetitions

$$\mathbf{I} \overset{\mathcal{D}}{=} \textit{Multinomial}(n-k, \mathbf{D})$$

Distribution of Partition Sizes I

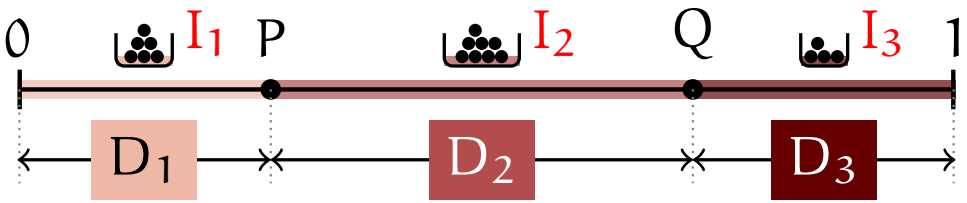
Recall: *Input* n ***i. i. d.*** *Uniform*$(0,1)$ *variables.*

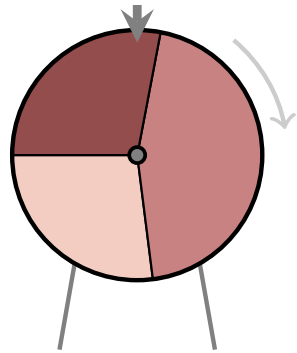
- Assume **fixed** **pivot values** P and Q

$\rightsquigarrow$ **Classes** of elements are **i. i. d.** with

$$\Pr[x \text{ small }] = P \quad =: D_1$$
$$\Pr[x \text{ medium }] = Q - P \quad =: D_2$$
$$\Pr[x \text{ large }] = 1 - Q \quad =: D_3$$

- **Nicer:** In terms of **Spacings** $\mathbf{D} = (D_1, D_2, D_3)$





$\rightsquigarrow$ Partition sizes $\mathbf{I}$ = cumulative result of independent repetitions

$$\mathbf{I} \overset{\mathcal{D}}{=} \textit{Multinomial}(n-k, \mathbf{D})$$

Distribution of Partition Sizes I

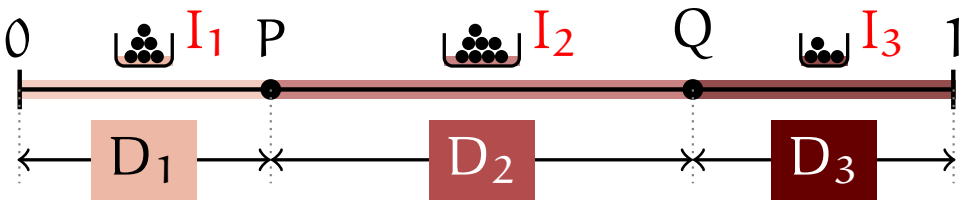
Recall: *Input* n ***i. i. d.*** *Uniform*$(0,1)$ *variables.*

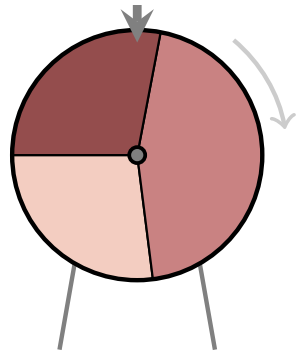
- Assume **fixed** **pivot values** P and Q

$\rightsquigarrow$ **Classes** of elements are **i. i. d.** with

$$\Pr[x \text{ small }] = P \quad =: D_1$$
$$\Pr[x \text{ medium }] = Q - P \quad =: D_2$$
$$\Pr[x \text{ large }] = 1 - Q \quad =: D_3$$

- **Nicer:** In terms of **Spacings** $\mathbf{D} = (D_1, D_2, D_3)$





$\rightsquigarrow$ Partition sizes $\mathbf{I}$ = cumulative result of independent repetitions

$$\mathbf{I} \overset{\mathcal{D}}{=} \textit{Multinomial}(n-k, \mathbf{D})$$

Distribution of Pivot *Values* P *and* Q

P and Q are ***order statistics*** of k **i. i. d.** ***Uniform***$(0,1)$ variables



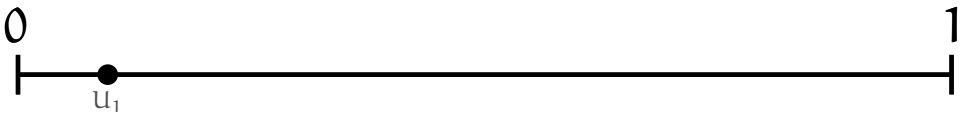
$\mathbf{t} = (1,2,3)$

$\rightsquigarrow$ Density for spacings: $f_{\mathbf{D}}(d_1,d_2,d_3) \propto d_1^{t_1} \cdot d_2^{t_2} \cdot d_3^{t_3}$

$\rightsquigarrow \mathbf{D} \overset{\mathcal{D}}{=} \mathit{Dirichlet}(\mathbf{t}+1) = \mathit{Dirichlet}(t_1+1, t_2+1, t_3+1)$

Distribution of Pivot *Values* P *and* Q

P and Q are ***order statistics*** of k **i. i. d.** ***Uniform***$(0,1)$ variables



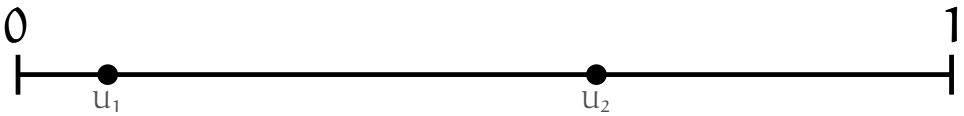
$\mathbf{t} = (1,2,3)$

$\rightsquigarrow$ Density for spacings: $f_{\mathbf{D}}(d_1,d_2,d_3) \;\propto\; d_1^{t_1} \cdot d_2^{t_2} \cdot d_3^{t_3}$

$\rightsquigarrow$ $\mathbf{D} \overset{\mathcal{D}}{=} \mathit{Dirichlet}(\mathbf{t}+1) = \mathit{Dirichlet}(t_1+1, t_2+1, t_3+1)$

Distribution of Pivot *Values* P *and* Q

P and Q are ***order statistics*** of k **i. i. d.** ***Uniform***$(0,1)$ variables



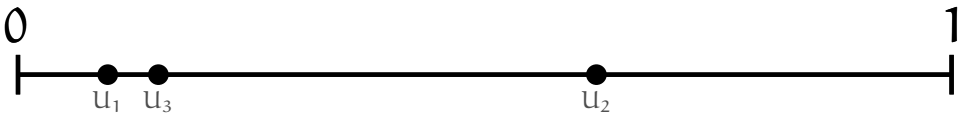
$\mathbf{t} = (1,2,3)$

$\rightsquigarrow$ Density for spacings: $f_{\mathbf{D}}(d_1,d_2,d_3) \;\propto\; d_1^{t_1} \cdot d_2^{t_2} \cdot d_3^{t_3}$

$\rightsquigarrow$ $\mathbf{D} \overset{\mathcal{D}}{=} \mathit{Dirichlet}(\mathbf{t}+1) = \mathit{Dirichlet}(t_1+1, t_2+1, t_3+1)$

Distribution of Pivot *Values* P *and* Q

P and Q are ***order statistics*** of k **i. i. d.** ***Uniform***$(0,1)$ variables



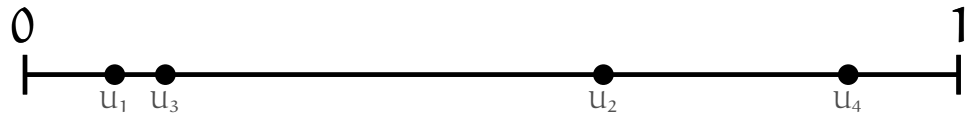
$\mathbf{t} = (1,2,3)$

$\rightsquigarrow$ Density for spacings: $f_{\mathbf{D}}(d_1,d_2,d_3) \;\propto\; d_1^{t_1} \cdot d_2^{t_2} \cdot d_3^{t_3}$

$\rightsquigarrow$ $\mathbf{D} \overset{\mathcal{D}}{=} \mathit{Dirichlet}(\mathbf{t}+1) = \mathit{Dirichlet}(t_1+1, t_2+1, t_3+1)$

Distribution of Pivot *Values* P *and* Q

P and Q are ***order statistics*** of k **i. i. d.** ***Uniform***$(0,1)$ variables



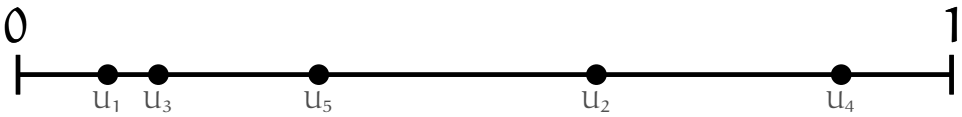
$\mathbf{t} = (1,2,3)$

$\rightsquigarrow$ Density for spacings: $f_{\mathbf{D}}(d_1,d_2,d_3) \;\propto\; d_1^{t_1} \cdot d_2^{t_2} \cdot d_3^{t_3}$

$\rightsquigarrow$ $\mathbf{D} \overset{\mathcal{D}}{=} \mathit{Dirichlet}(\mathbf{t}+1) = \mathit{Dirichlet}(t_1+1, t_2+1, t_3+1)$

Distribution of Pivot *Values* P *and* Q

P and Q are ***order statistics*** of k **i. i. d.** ***Uniform***$(0,1)$ variables



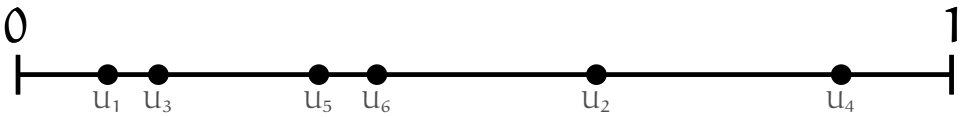
$\mathbf{t} = (1,2,3)$

$\rightsquigarrow$ Density for spacings: $f_{\mathbf{D}}(d_1,d_2,d_3) \;\propto\; d_1^{t_1} \cdot d_2^{t_2} \cdot d_3^{t_3}$

$\rightsquigarrow$ $\mathbf{D} \overset{\mathcal{D}}{=} \mathit{Dirichlet}(\mathbf{t}+1) = \mathit{Dirichlet}(t_1+1, t_2+1, t_3+1)$

Distribution of Pivot *Values* P *and* Q

P and Q are ***order statistics*** of k **i. i. d.** ***Uniform***$(0,1)$ variables



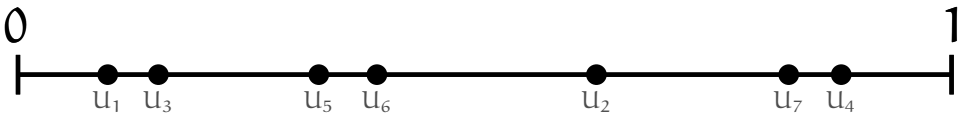
$\mathbf{t} = (1,2,3)$

$\rightsquigarrow$ Density for spacings: $f_{\mathbf{D}}(d_1,d_2,d_3) \;\propto\; d_1^{t_1} \cdot d_2^{t_2} \cdot d_3^{t_3}$

$\rightsquigarrow$ $\mathbf{D} \overset{\mathcal{D}}{=} \mathit{Dirichlet}(\mathbf{t}+1) = \mathit{Dirichlet}(t_1+1, t_2+1, t_3+1)$

Distribution of Pivot *Values* P *and* Q

P and Q are ***order statistics*** of k **i. i. d.** ***Uniform***$(0,1)$ variables



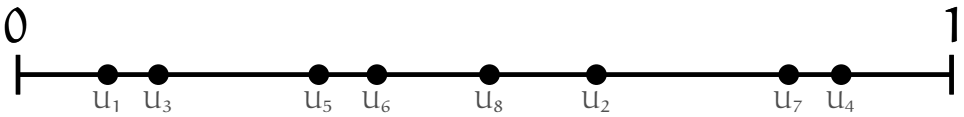
$\mathbf{t} = (1,2,3)$

$\rightsquigarrow$ Density for spacings: $f_{\mathbf{D}}(d_1,d_2,d_3) \propto d_1^{t_1} \cdot d_2^{t_2} \cdot d_3^{t_3}$

$\rightsquigarrow \mathbf{D} \overset{\mathcal{D}}{=} \mathit{Dirichlet}(\mathbf{t}+1) = \mathit{Dirichlet}(t_1+1, t_2+1, t_3+1)$

Distribution of Pivot *Values* P *and* Q

P and Q are ***order statistics*** of k **i. i. d.** ***Uniform***$(0,1)$ variables



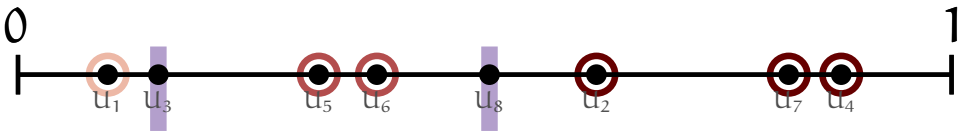
$\mathbf{t} = (1,2,3)$

$\rightsquigarrow$ Density for spacings: $f_{\mathbf{D}}(d_1,d_2,d_3) \propto d_1^{t_1} \cdot d_2^{t_2} \cdot d_3^{t_3}$

$\rightsquigarrow \mathbf{D} \overset{\mathcal{D}}{=} \mathit{Dirichlet}(\mathbf{t}+1) = \mathit{Dirichlet}(t_1+1, t_2+1, t_3+1)$

Distribution of Pivot *Values* P *and* Q

P and Q are ***order statistics*** of k **i. i. d.** ***Uniform***$(0,1)$ variables



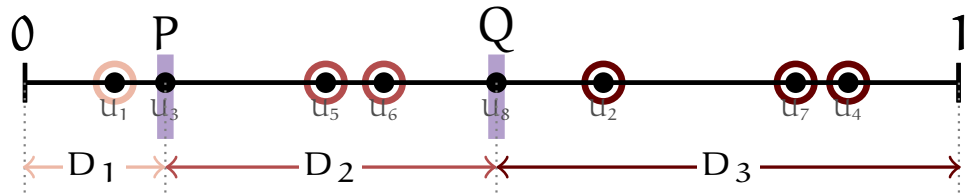
$\mathbf{t} = (1,2,3)$

$\rightsquigarrow$ Density for spacings: $f_{\mathbf{D}}(d_1,d_2,d_3) \;\propto\; d_1^{t_1} \cdot d_2^{t_2} \cdot d_3^{t_3}$

$\rightsquigarrow$ $\mathbf{D} \overset{\mathcal{D}}{=} \mathit{Dirichlet}(\mathbf{t}+1) = \mathit{Dirichlet}(t_1+1, t_2+1, t_3+1)$

Distribution of Pivot *Values* P *and* Q

P and Q are ***order statistics*** of k **i. i. d.** ***Uniform***$(0,1)$ variables



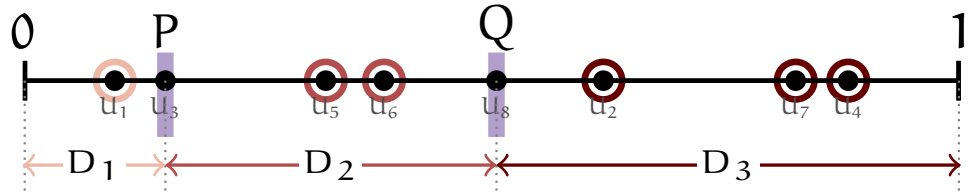
$\mathbf{t} = (1,2,3)$

$\rightsquigarrow$ Density for spacings: $f_{\mathbf{D}}(d_1,d_2,d_3) \propto d_1^{t_1} \cdot d_2^{t_2} \cdot d_3^{t_3}$

$\rightsquigarrow$ $\mathbf{D} \overset{\mathcal{D}}{=} \mathit{Dirichlet}(\mathbf{t}+1) = \mathit{Dirichlet}(t_1+1, t_2+1, t_3+1)$

Distribution of Pivot *Values* P *and* Q

P and Q are ***order statistics*** of k **i. i. d.** ***Uniform***$(0,1)$ variables



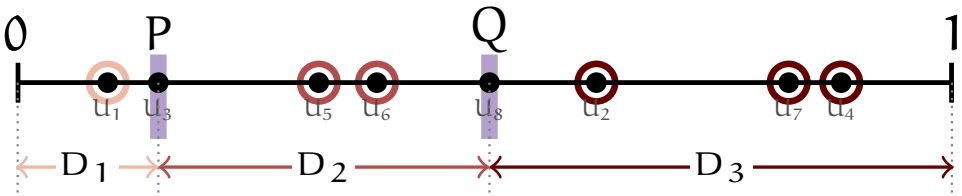
$\mathbf{t} = (1, 2, 3)$

$\rightsquigarrow$ Density for spacings: $f_{\mathbf{D}}(d_1, d_2, d_3) \propto d_1^{t_1} \cdot d_2^{t_2} \cdot d_3^{t_3}$

$\rightsquigarrow \mathbf{D} \overset{\mathcal{D}}{=} \mathit{Dirichlet}(\mathbf{t}+1) = \mathit{Dirichlet}(t_1+1, t_2+1, t_3+1)$

Distribution of Pivot *Values* P *and* Q

P and Q are ***order statistics*** of k **i. i. d.** ***Uniform***$(0,1)$ variables



$\mathbf{t} = (1,2,3)$

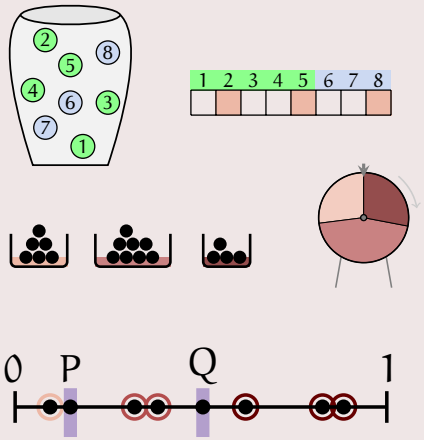
$\rightsquigarrow$ Density for spacings: $f_{\mathbf{D}}(d_1,d_2,d_3) \quad \propto \quad d_1^{t_1} \cdot d_2^{t_2} \cdot d_3^{t_3}$

$\rightsquigarrow$ $\mathbf{D} \overset{\mathcal{D}}{=} \textit{Dirichlet}(\mathbf{t}+1) = \textit{Dirichlet}(t_1+1, t_2+1, t_3+1)$

Expected #Comparisons per Partitioning

Summary of distributional analysis

- $s@\mathcal{K} \overset{\mathcal{D}}{=} \mathit{HypG}(I_1, I_1 + I_2, n) + O(1)$
- $\mathbf{I} \overset{\mathcal{D}}{=} \mathit{Mult}(n, \mathbf{D}) + O(1)$
- $\mathbf{D} \overset{\mathcal{D}}{=} \mathit{Dir}(\mathbf{t} + 1)$

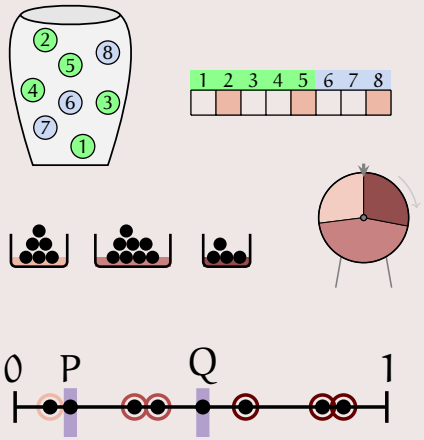


⇝ *expected value by* ***successive unconditioning:***

Expected #Comparisons per Partitioning

Summary of distributional analysis

- $s@\mathcal{K} \overset{\mathcal{D}}{=} \mathit{HypG}(I_1, I_1 + I_2, n) + O(1)$
- $\mathbf{I} \overset{\mathcal{D}}{=} \mathit{Mult}(n, \mathbf{D}) + O(1)$
- $\mathbf{D} \overset{\mathcal{D}}{=} \mathit{Dir}(\mathbf{t} + 1)$



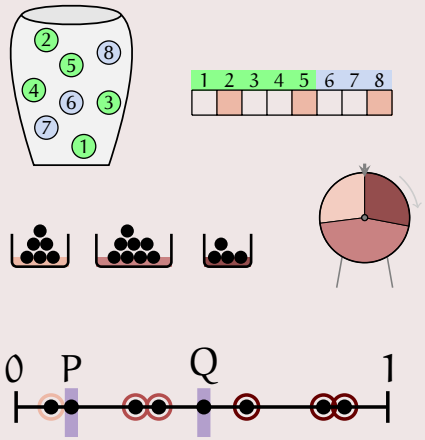
$\rightsquigarrow$ *expected value by* ***successive unconditioning:***

$$\mathbb{E}[\, s@\mathcal{K} \quad]$$

Expected #Comparisons per Partitioning

Summary of distributional analysis

- $s@\mathcal{K} \overset{\mathcal{D}}{=} \mathit{HypG}(I_1, I_1 + I_2, n) + O(1)$
- $\mathbf{I} \overset{\mathcal{D}}{=} \mathit{Mult}(n, \mathbf{D}) + O(1)$
- $\mathbf{D} \overset{\mathcal{D}}{=} \mathit{Dir}(\mathbf{t} + 1)$



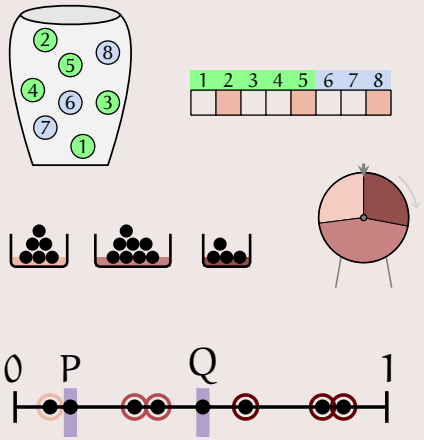
$\rightsquigarrow$ *expected value by* ***successive unconditioning:***

$$\mathbb{E}[s@\mathcal{K} \mid \mathbf{I}]$$

Expected #Comparisons per Partitioning

Summary of distributional analysis

- $s@\mathcal{K} \overset{\mathcal{D}}{=} \mathit{HypG}(I_1, I_1 + I_2, n) + O(1)$
- $\mathbf{I} \overset{\mathcal{D}}{=} \mathit{Mult}(n, \mathbf{D}) + O(1)$
- $\mathbf{D} \overset{\mathcal{D}}{=} \mathit{Dir}(\mathbf{t} + 1)$



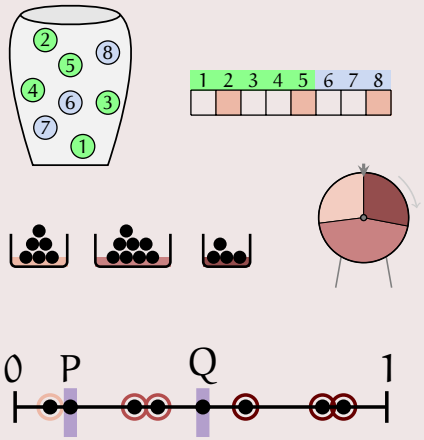
$\rightsquigarrow$ *expected value by* ***successive unconditioning:***

$$\mathbb{E}_{\mathbf{I}}\left[\mathbb{E}[s@\mathcal{K} \mid \mathbf{I}] \,\middle|\, \mathbf{D}\right]$$

Expected #Comparisons per Partitioning

Summary of distributional analysis

- $s@\mathcal{K} \overset{\mathcal{D}}{=} \mathit{HypG}(I_1, I_1 + I_2, n) + O(1)$
- $\mathbf{I} \overset{\mathcal{D}}{=} \mathit{Mult}(n, \mathbf{D}) + O(1)$
- $\mathbf{D} \overset{\mathcal{D}}{=} \mathit{Dir}(\mathbf{t} + 1)$



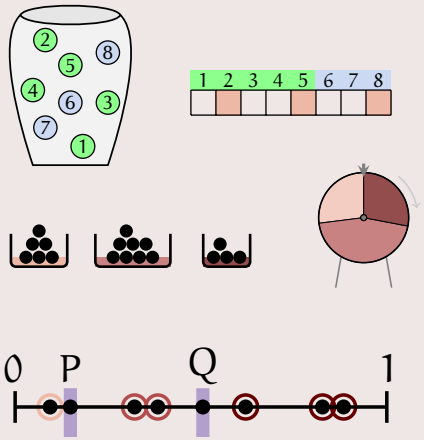
$\rightsquigarrow$ *expected value by* ***successive unconditioning:***

$$\mathbb{E}_{\mathbf{D}}\Big[\mathbb{E}_{\mathbf{I}}\big[\mathbb{E}[s@\mathcal{K} \mid \mathbf{I}] \,\big|\, \mathbf{D}\big]\Big]$$

Expected #Comparisons per Partitioning

Summary of distributional analysis

- $s@\mathcal{K} \overset{\mathcal{D}}{=} \mathit{HypG}(I_1, I_1 + I_2, n) + O(1)$
- $\mathbf{I} \overset{\mathcal{D}}{=} \mathit{Mult}(n, \mathbf{D}) + O(1)$
- $\mathbf{D} \overset{\mathcal{D}}{=} \mathit{Dir}(\mathbf{t} + 1)$



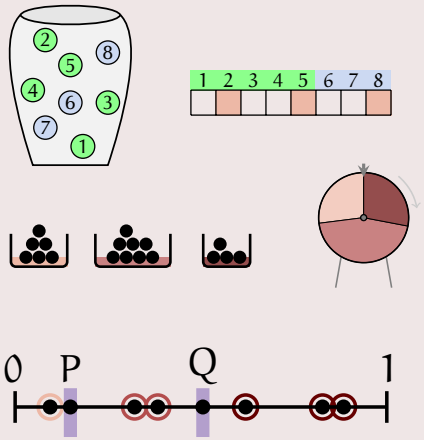
$\rightsquigarrow$ *expected value by* ***successive unconditioning:***

$$\mathbb{E}_{\mathbf{D}}\Big[\mathbb{E}_{\mathbf{I}}\big[\mathbb{E}[s@\mathcal{K} \mid \mathbf{I}] \,\big|\, \mathbf{D}\big]\Big] = \frac{(t_1+1)(t_1+t_2+3)}{(k+2)(k+1)} \cdot n + O(1)$$

Expected #Comparisons per Partitioning

Summary of distributional analysis

- $s@\mathcal{K} \overset{\mathcal{D}}{=} HypG(I_1, I_1 + I_2, n) + O(1)$
- $\mathbf{I} \overset{\mathcal{D}}{=} Mult(n, \mathbf{D}) + O(1)$
- $\mathbf{D} \overset{\mathcal{D}}{=} Dir(\mathbf{t} + 1)$



$\rightsquigarrow$ *expected value by* ***successive unconditioning:***

$$\mathbb{E}_{\mathbf{D}}\Big[\mathbb{E}_{\mathbf{I}}\big[\mathbb{E}[s@\mathcal{K} \mid \mathbf{I}] \,\big|\, \mathbf{D}\big]\Big] = \frac{(t_1+1)(t_1+t_2+3)}{(k+2)(k+1)} \cdot n + O(1)$$

$$\mathbb{E}_{\mathbf{D}}\Big[\mathbb{E}_{\mathbf{I}}\big[\mathbb{E}[l@\mathcal{G} \mid \mathbf{I}] \,\big|\, \mathbf{D}\big]\Big] = \frac{(t_3+2)(t_3+1)}{(k+2)(k+1)} \cdot n + O(1)$$

Expected #Comparisons

Final Result:

$$\mathbb{E}[C_n] \sim \frac{\gamma}{H} \cdot n \ln n$$

Expected #Comparisons

Final Result:

$$\mathbb{E}[C_n] \sim \frac{\gamma}{\sum_{l=1}^{3} \frac{t_l + 1}{k+1} \left(\mathcal{H}_{k+1} - \mathcal{H}_{t_l+1} \right)} \cdot n \ln n$$

Expected #Comparisons

Final Result:

$$\mathbb{E}[C_n] \sim \frac{2 - \frac{s\,@\,\mathcal{K}}{n} - \frac{l\,@\,\mathcal{G}}{n}}{\sum_{l=1}^{3} \frac{t_l + 1}{k+1}\left(\mathcal{H}_{k+1} - \mathcal{H}_{t_l+1}\right)} \cdot n \ln n$$

Expected #Comparisons

Final Result:

$$\mathbb{E}[C_n] \sim \frac{2 - \frac{(t_1+1)(t_1+t_2+3)}{(k+2)(k+1)} - \frac{(t_3+2)(t_3+1)}{(k+2)(k+1)}}{\sum_{l=1}^{3} \frac{t_l+1}{k+1}\left(\mathcal{H}_{k+1} - \mathcal{H}_{t_l+1}\right)} \cdot n \ln n$$

Expected #Comparisons

Final Result:

$$\mathbb{E}[C_n] \sim \frac{2 - \frac{(t_1+1)(t_1+t_2+3)}{(k+2)(k+1)} - \frac{(t_3+2)(t_3+1)}{(k+2)(k+1)}}{\sum_{\iota=1}^{3} \frac{t_\iota+1}{k+1}\left(\mathcal{H}_{k+1} - \mathcal{H}_{t_\iota+1}\right)} \cdot n \ln n$$

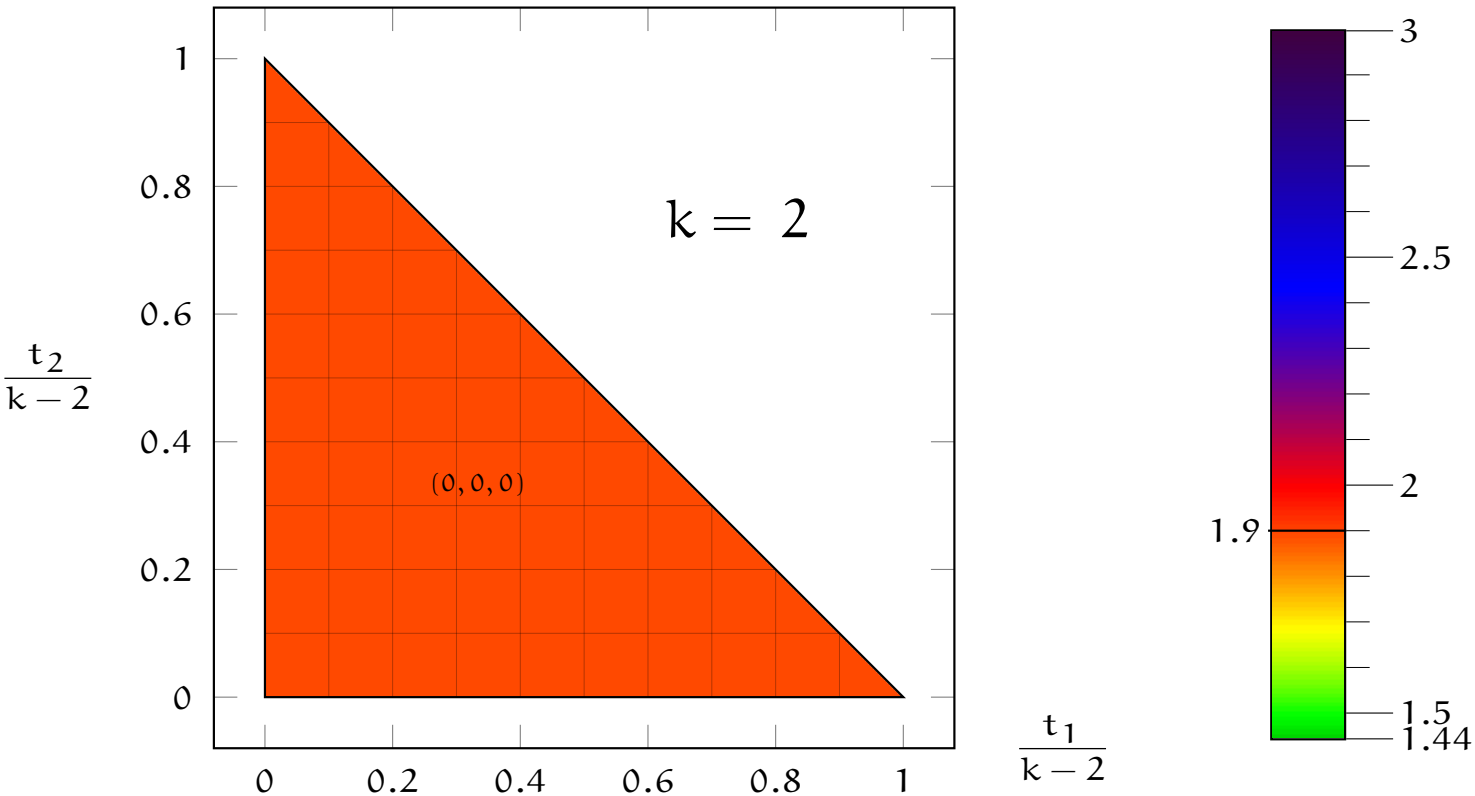
. . . a little hard to interpret.

Let's have some pictures.

Expected #Comparisons

Final Result:

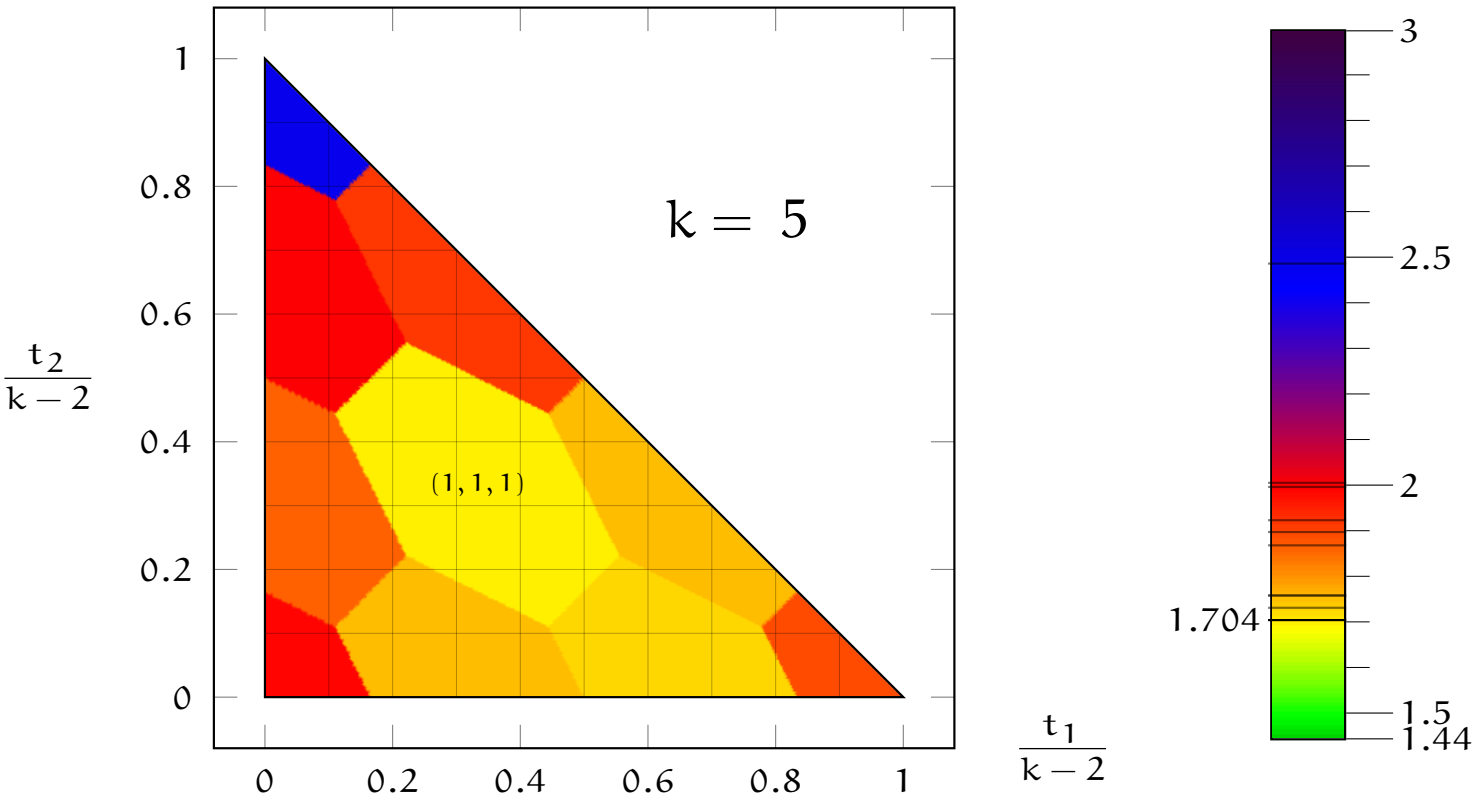
$$\mathbb{E}[C_n] \sim \frac{2 - \frac{(t_1+1)(t_1+t_2+3)}{(k+2)(k+1)} - \frac{(t_3+2)(t_3+1)}{(k+2)(k+1)}}{\sum_{l=1}^{3} \frac{t_l+1}{k+1}\left(\mathcal{H}_{k+1} - \mathcal{H}_{t_l+1}\right)} \cdot n \ln n$$



Expected #Comparisons

Final Result:

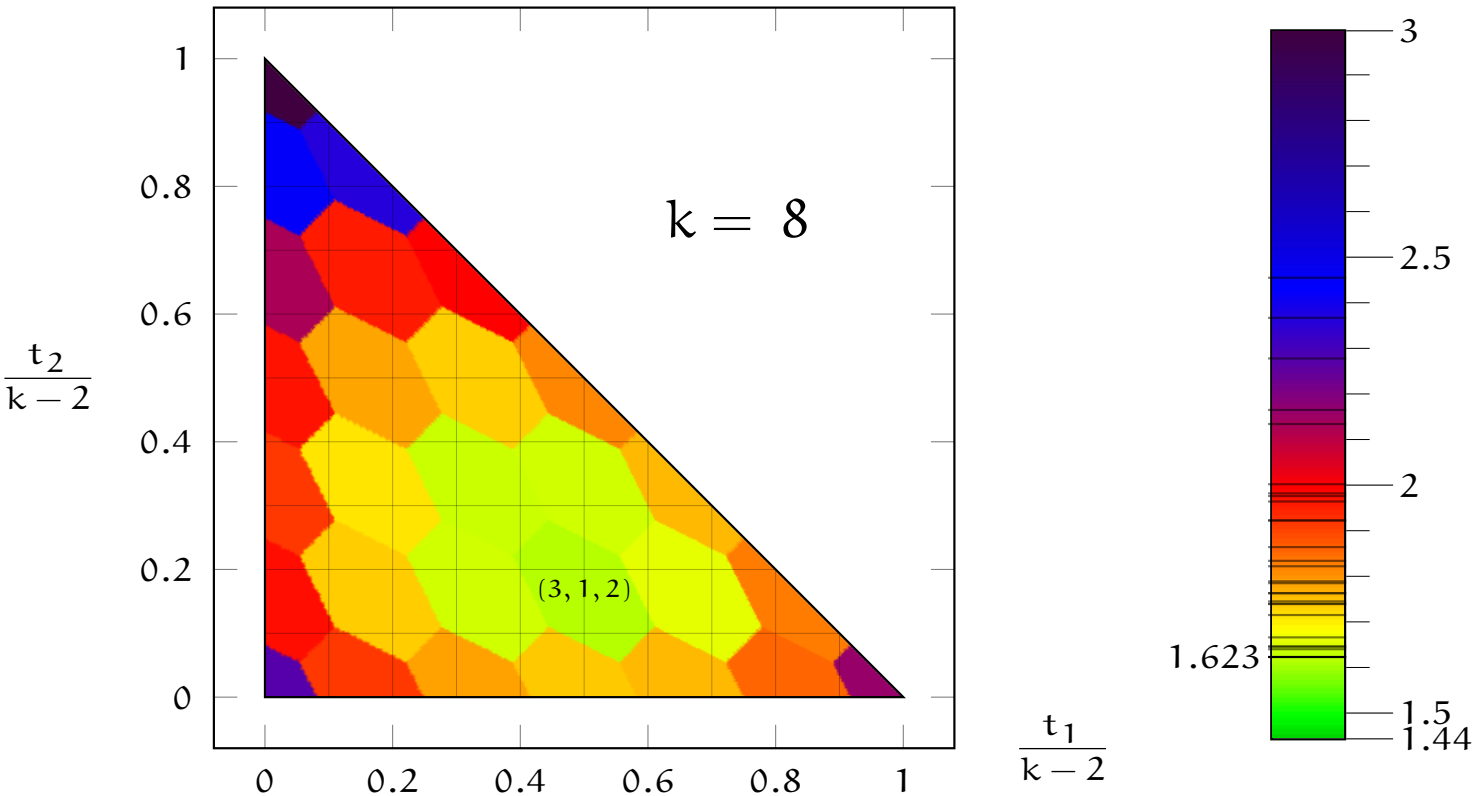
$$\mathbb{E}[C_n] \sim \frac{2 - \dfrac{(t_1+1)(t_1+t_2+3)}{(k+2)(k+1)} - \dfrac{(t_3+2)(t_3+1)}{(k+2)(k+1)}}{\displaystyle\sum_{l=1}^{3} \frac{t_l+1}{k+1}\left(\mathcal{H}_{k+1} - \mathcal{H}_{t_l+1}\right)} \cdot n \ln n$$



Expected #Comparisons

Final Result:

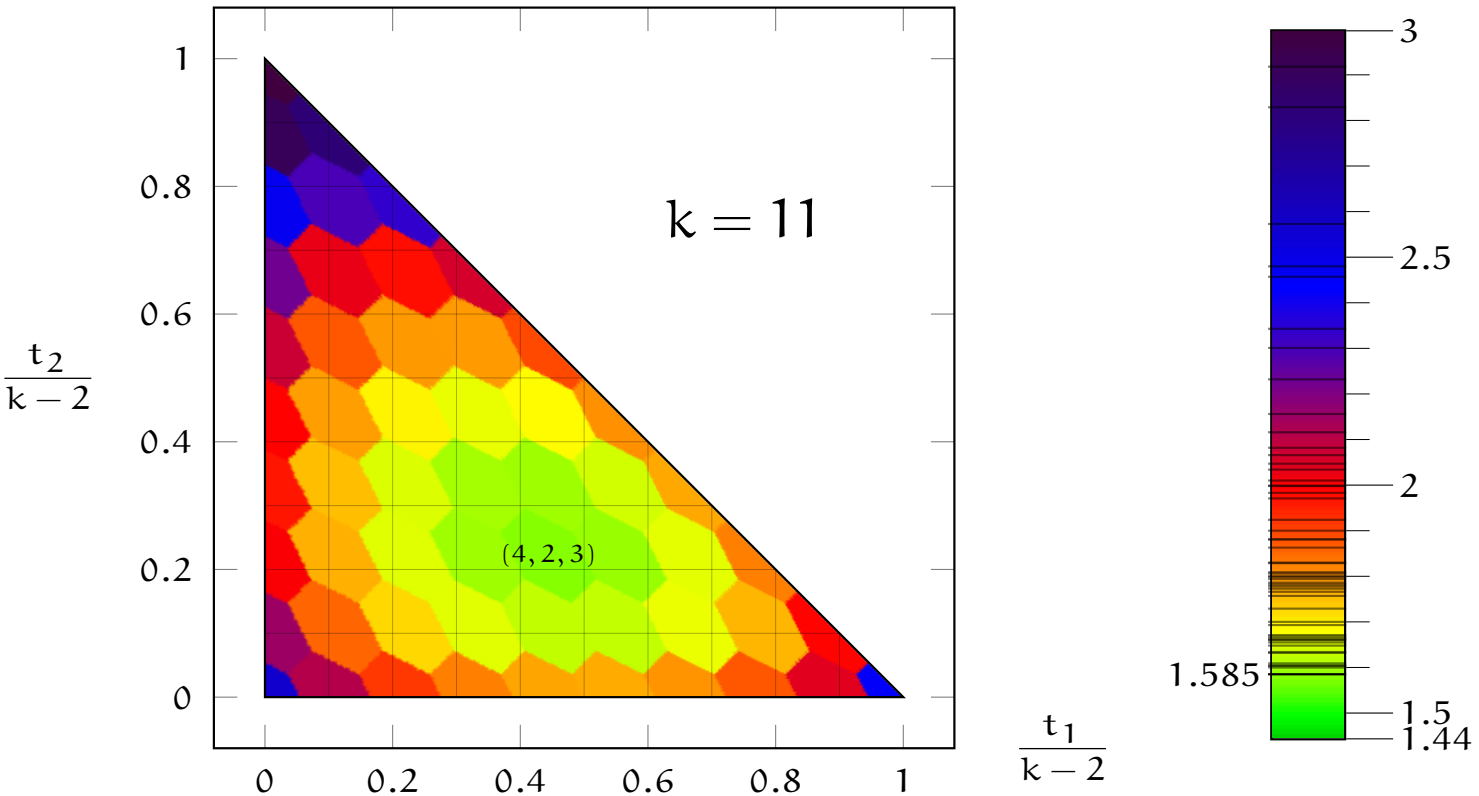
$$\mathbb{E}[C_n] \sim \frac{2 - \frac{(t_1+1)(t_1+t_2+3)}{(k+2)(k+1)} - \frac{(t_3+2)(t_3+1)}{(k+2)(k+1)}}{\sum_{l=1}^{3} \frac{t_l+1}{k+1}\left(\mathcal{H}_{k+1} - \mathcal{H}_{t_l+1}\right)} \cdot n \ln n$$



Expected #Comparisons

Final Result:

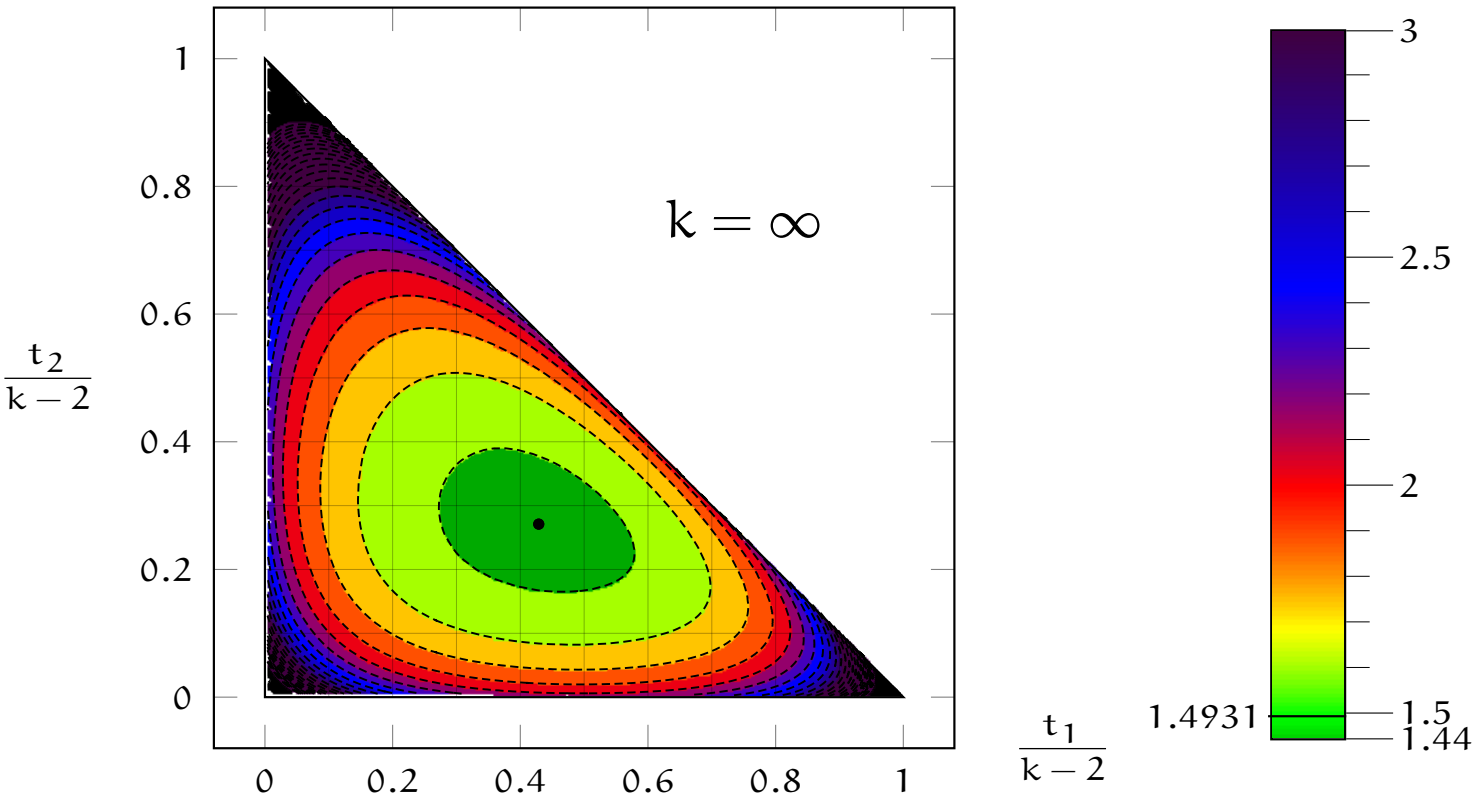
$$\mathbb{E}[C_n] \sim \frac{2 - \dfrac{(t_1+1)(t_1+t_2+3)}{(k+2)(k+1)} - \dfrac{(t_3+2)(t_3+1)}{(k+2)(k+1)}}{\displaystyle\sum_{l=1}^{3} \frac{t_l+1}{k+1}\left(\mathcal{H}_{k+1} - \mathcal{H}_{t_l+1}\right)} \cdot n \ln n$$



Expected #Comparisons

Final Result:

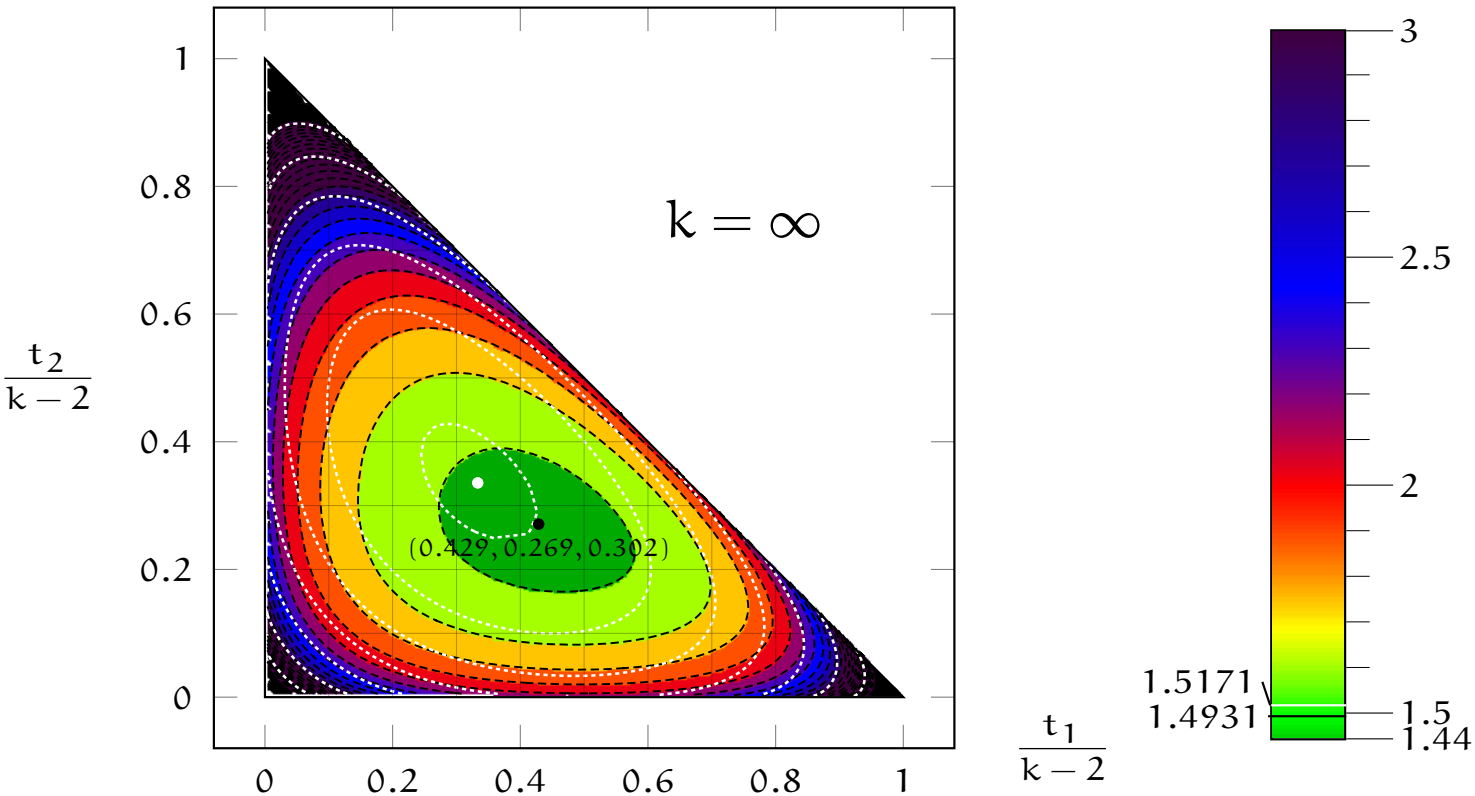
$$\mathbb{E}[C_n] \sim \frac{2 - \dfrac{(t_1+1)(t_1+t_2+3)}{(k+2)(k+1)} - \dfrac{(t_3+2)(t_3+1)}{(k+2)(k+1)}}{\displaystyle\sum_{l=1}^{3} \frac{t_l+1}{k+1}\left(\mathcal{H}_{k+1} - \mathcal{H}_{t_l+1}\right)} \cdot n \ln n$$



Expected #Comparisons

Final Result:

$$\mathbb{E}[C_n] \sim \frac{2 - \dfrac{(t_1+1)(t_1+t_2+3)}{(k+2)(k+1)} - \dfrac{(t_3+2)(t_3+1)}{(k+2)(k+1)}}{\displaystyle\sum_{l=1}^{3} \frac{t_l+1}{k+1}\left(\mathcal{H}_{k+1} - \mathcal{H}_{t_l+1}\right)} \cdot n \ln n$$



Beyond Comparisons

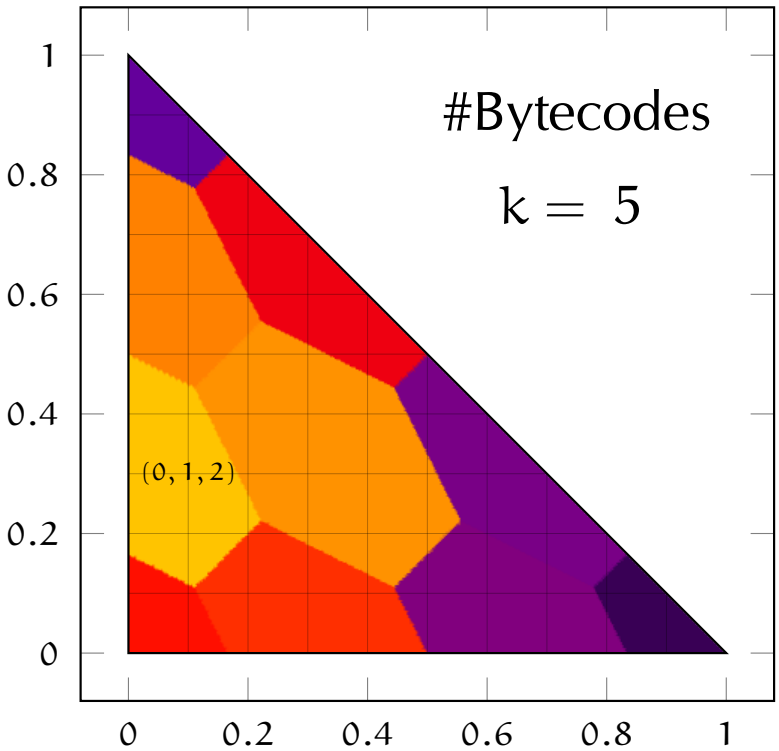
Similarly: *full (basic block) frequency analysis*

⇝ #Bytecode instructions, MMIX costs etc.

Beyond Comparisons

Similarly: *full (basic block) frequency analysis*

⇝ #Bytecode instructions, MMIX costs etc.



- Optimum for **Bytecodes**
  $\tau^*_{BC} \approx (0.2068, 0.3485, 0.4447)$

- **Recall:**
  Optimum for **comparisons**
  $\tau^* \approx (0.4288, 0.2688, 0.3024)$

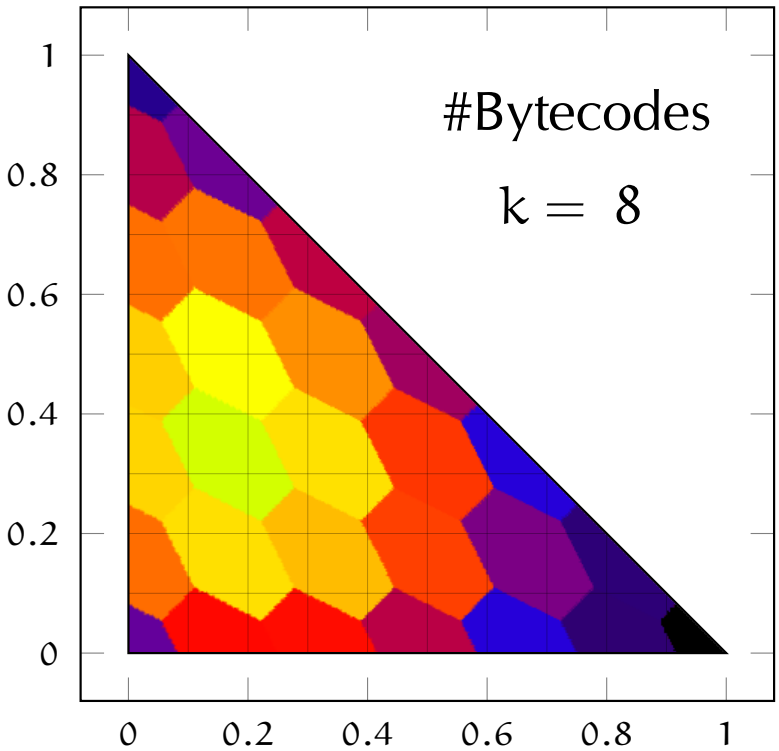
⇝ **Contrary skew** of pivots needed!

⇝ *Optimal skew of pivots heavily depends on **cost measure**!*

Beyond Comparisons

Similarly: *full (basic block) frequency analysis*

⇝ #Bytecode instructions, MMIX costs etc.



- Optimum for **Bytecodes**
  $\tau^*_{BC} \approx (0.2068, 0.3485, 0.4447)$

- **Recall:**
  Optimum for **comparisons**
  $\tau^* \approx (0.4288, 0.2688, 0.3024)$

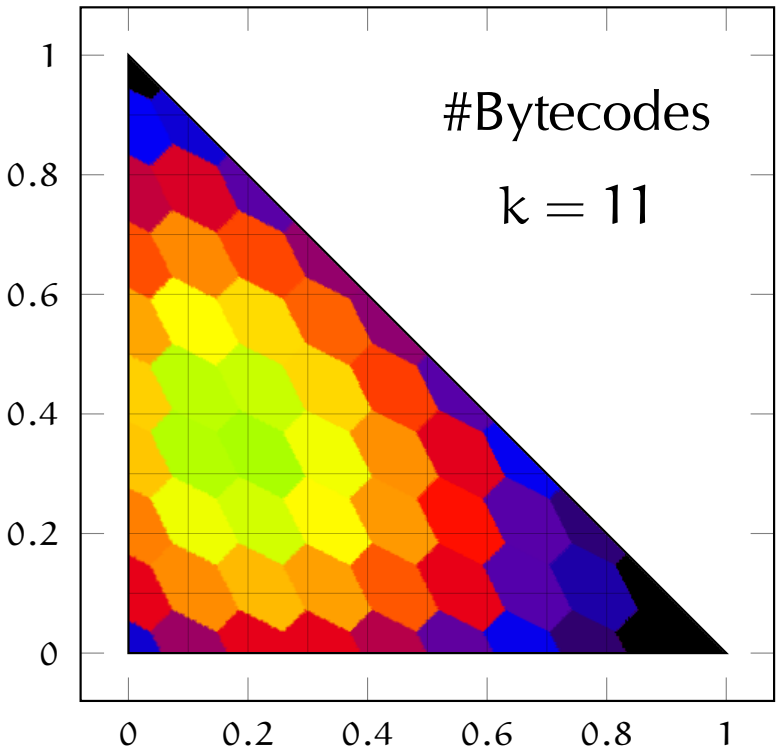
⇝ **Contrary skew** of pivots needed!

⇝ *Optimal skew of pivots heavily depends on **cost measure**!*

Beyond Comparisons

Similarly: *full (basic block) frequency analysis*

⇝ #Bytecode instructions, MMIX costs etc.



- Optimum for **Bytecodes**
  $\tau^*_{BC} \approx (0.2068, 0.3485, 0.4447)$

- **Recall:**
  Optimum for **comparisons**
  $\tau^* \approx (0.4288, 0.2688, 0.3024)$

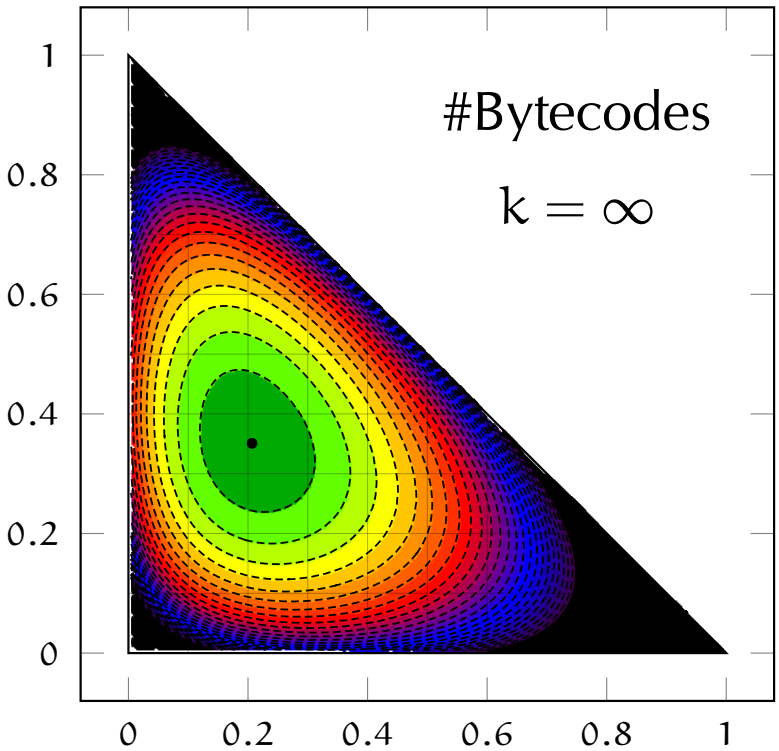
⇝ **Contrary skew** of pivots needed!

⇝ *Optimal skew of pivots heavily depends on **cost measure**!*

Beyond Comparisons

Similarly: *full (basic block) frequency analysis*

⇝ #Bytecode instructions, MMIX costs etc.



- Optimum for **Bytecodes**
  $\tau^*_{BC} \approx (0.2068, 0.3485, 0.4447)$

- **Recall:**
  Optimum for **comparisons**
  $\tau^* \approx (0.4288, 0.2688, 0.3024)$

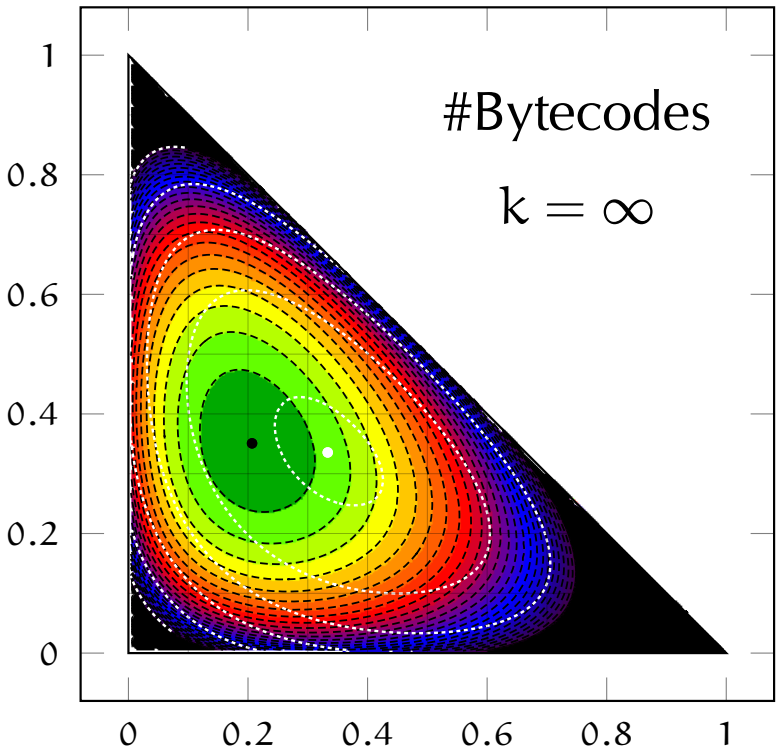
⇝ **Contrary skew** of pivots needed!

⇝ *Optimal skew of pivots heavily depends on **cost measure**!*

Beyond Comparisons

Similarly: *full (basic block) frequency analysis*

⇝ #Bytecode instructions, MMIX costs etc.



- Optimum for **Bytecodes**
  $\tau^*_{\text{BC}} \approx (0.2068, 0.3485, 0.4447)$



- **Recall:**
  Optimum for **comparisons**
  $\tau^* \approx (0.4288, 0.2688, 0.3024)$

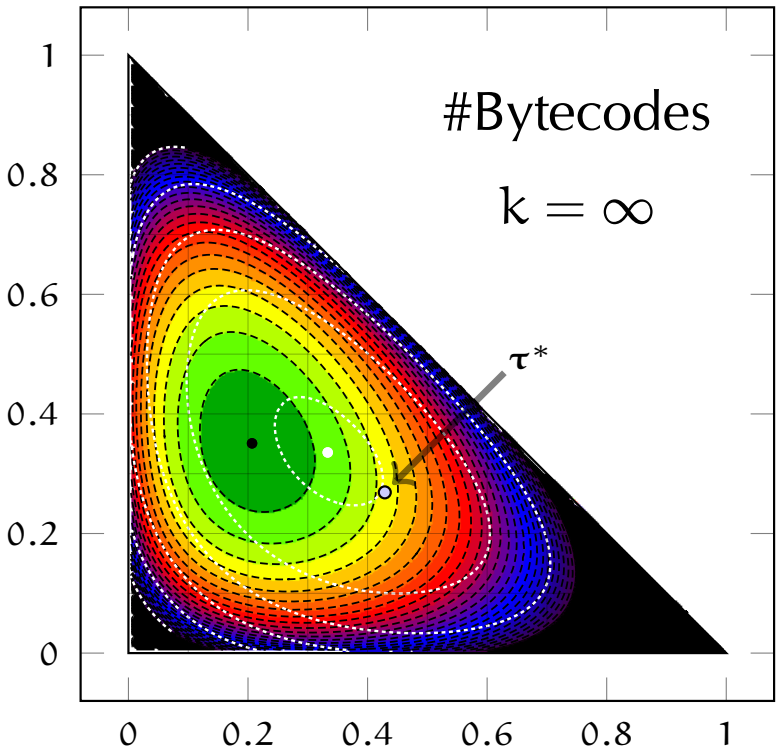
⇝ **Contrary skew** of pivots needed!

⇝ *Optimal skew of pivots heavily depends on **cost measure**!*

Beyond Comparisons

Similarly: *full (basic block) frequency analysis*

⇝ #Bytecode instructions, MMIX costs etc.



- Optimum for **Bytecodes**
  $\boldsymbol{\tau}^*_{\mathrm{BC}} \approx (0.2068, 0.3485, 0.4447)$



- **Recall:**
  Optimum for **comparisons**
  $\boldsymbol{\tau}^* \approx (0.4288, 0.2688, 0.3024)$



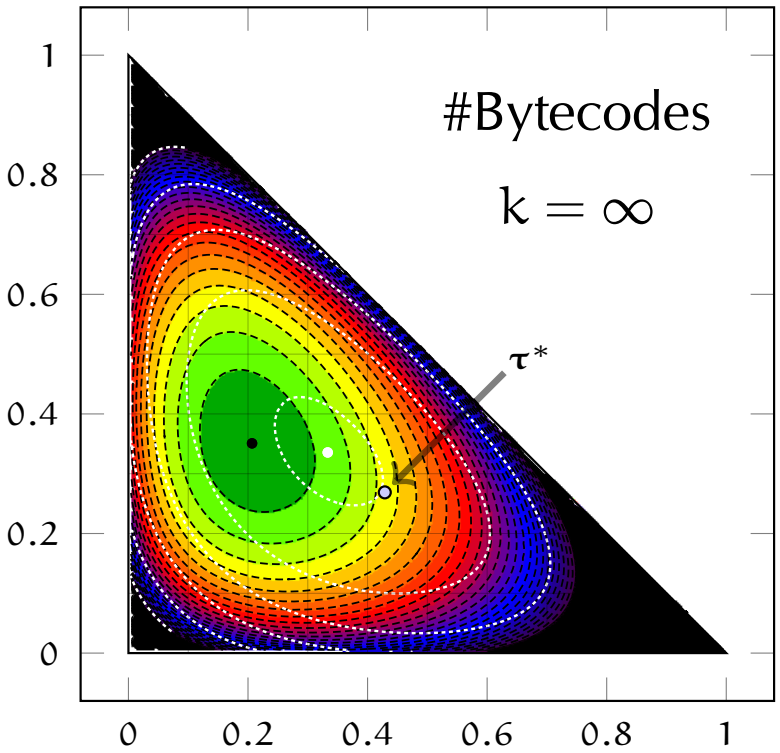
⇝ **Contrary skew** of pivots needed!

⇝ *Optimal skew of pivots heavily depends on* ***cost measure****!*

Beyond Comparisons

Similarly: *full (basic block) frequency analysis*

⇝ #Bytecode instructions, MMIX costs etc.



- Optimum for **Bytecodes**
  $\boldsymbol{\tau}^*_{\mathrm{BC}} \approx (0.2068, 0.3485, 0.4447)$



- **Recall:**
  Optimum for **comparisons**
  $\boldsymbol{\tau}^* \approx (0.4288, 0.2688, 0.3024)$



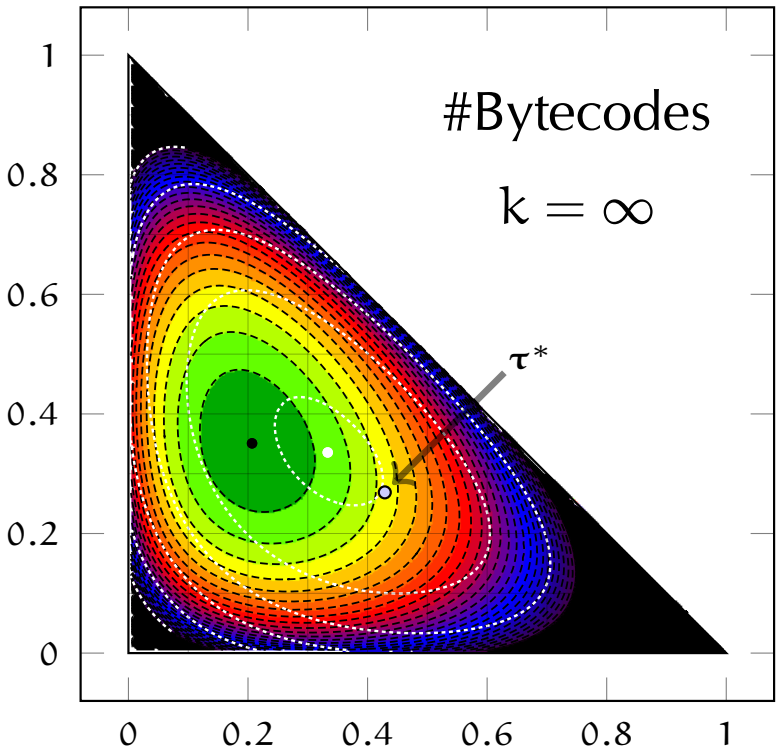
⇝ **Contrary skew** of pivots needed!

⇝ *Optimal skew of pivots heavily depends on* ***cost measure****!*

Beyond Comparisons

Similarly: *full (basic block) frequency analysis*

⇝ #Bytecode instructions, MMIX costs etc.



- Optimum for **Bytecodes**
  $\boldsymbol{\tau}^*_{\mathrm{BC}} \approx (0.2068, 0.3485, 0.4447)$



- **Recall:**
  Optimum for **comparisons**
  $\boldsymbol{\tau}^* \approx (0.4288, 0.2688, 0.3024)$



⇝ **Contrary skew** of pivots needed!

⇝ *Optimal skew of pivots heavily depends on* ***cost measure****!*

Conclusion

We've seen:

- **Asymmetric** partitioning calls for **skewed** pivots
- **Optimal** skew **sensitive** to employed cost measure

⇝ Realistic cost measure and detailed analysis needed!



Open Questions:

- What **exactly** makes Yaroslavskiy's algorithm fast in **practice**?
  Memory Hierarchy Effects?
- Is Yaroslavskiy's algorithm *special*?
  Are there even faster methods?
- Will they be asymmetric?

Conclusion

We've seen:

- **Asymmetric** partitioning calls for **skewed** pivots
- **Optimal** skew **sensitive** to employed cost measure

$\rightsquigarrow$ Realistic cost measure and detailed analysis needed!



Open Questions:

- What **exactly** makes Yaroslavskiy's algorithm fast in **practice**?
  Memory Hierarchy Effects?
- Is Yaroslavskiy's algorithm *special*?
  Are there even faster methods?
- Will they be asymmetric?