Precise Analysis of the Optimal Multi-Pivot Quicksort

Daniel Krenn

(joint work in progress with *Clemens Heuberger*)

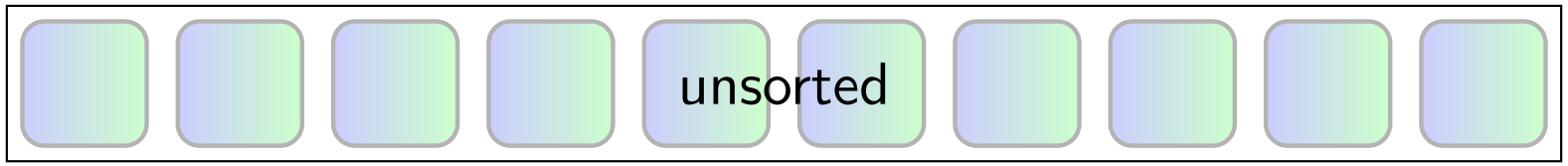
ALPEN-ADRIA UNIVERSITÄT
KLAGENFURT | WIEN GRAZ

June 19, 2017

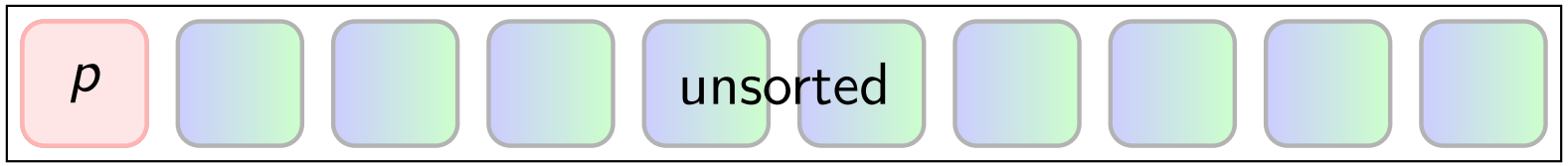
FWF Der Wissenschaftsfonds.

Supported by the Austrian Science Fund (FWF), project P24644.

Quicksort & Quickselect

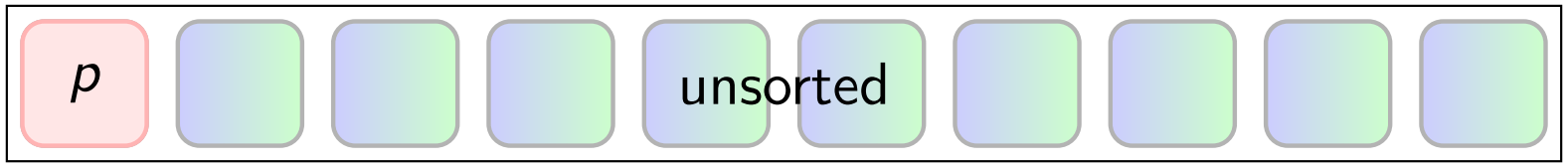


Quicksort & Quickselect

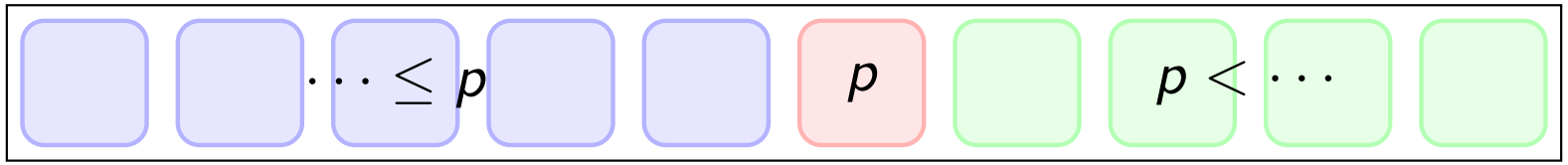


- choose a pivot element p

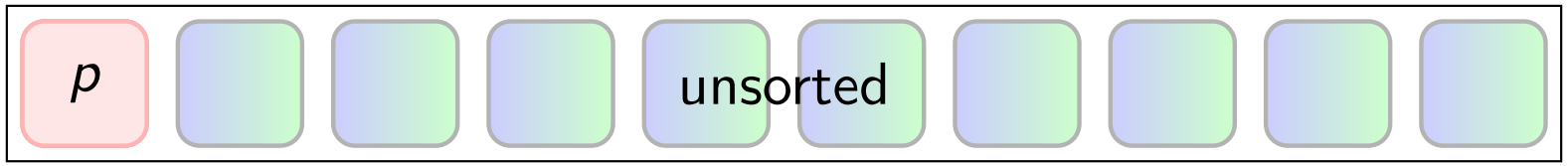
Quicksort & Quickselect



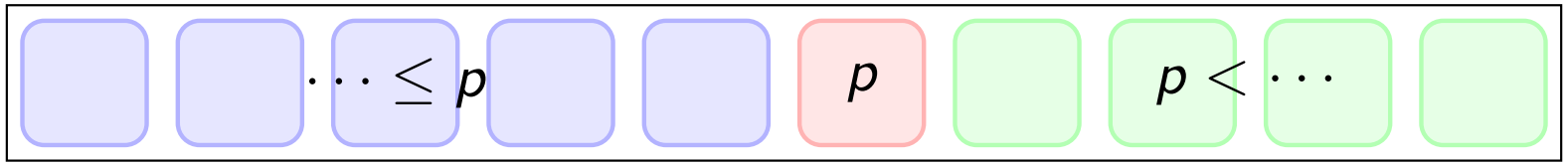
- choose a pivot element p
- partition into
  - small elements
  - large elements



Quicksort & Quickselect

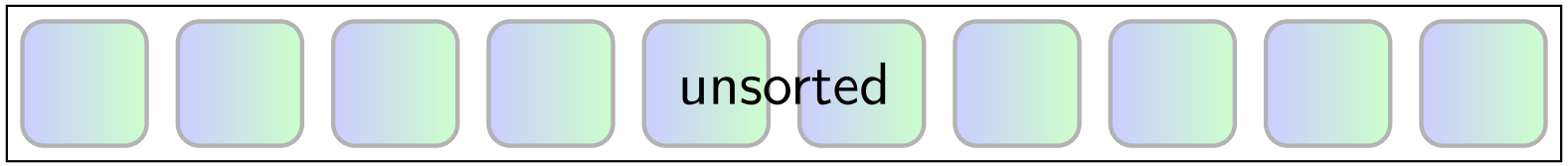


- choose a pivot element p
- partition into
  - small elements
  - large elements

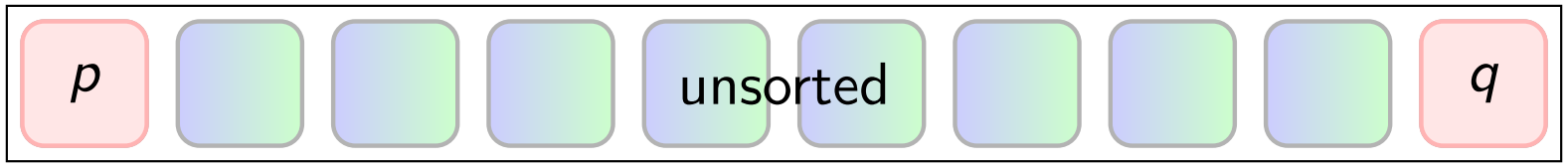


- proceed recursively

Dual Pivot Quicksort & Quickselect

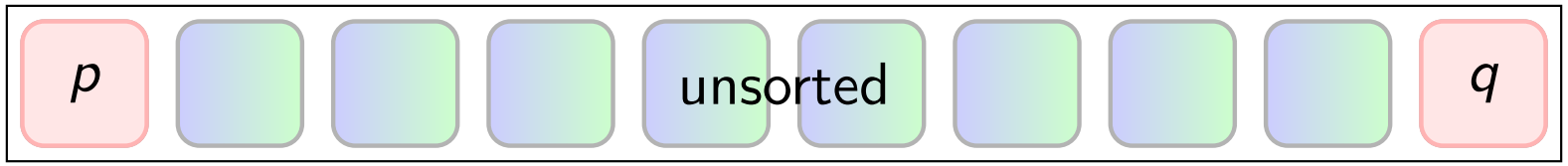


Dual Pivot Quicksort & Quickselect

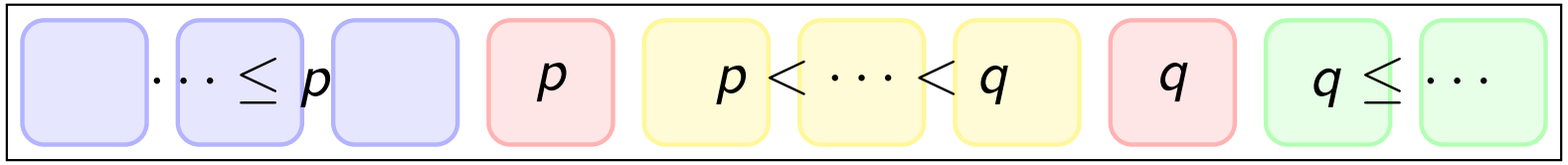


- choose pivot elements p and q

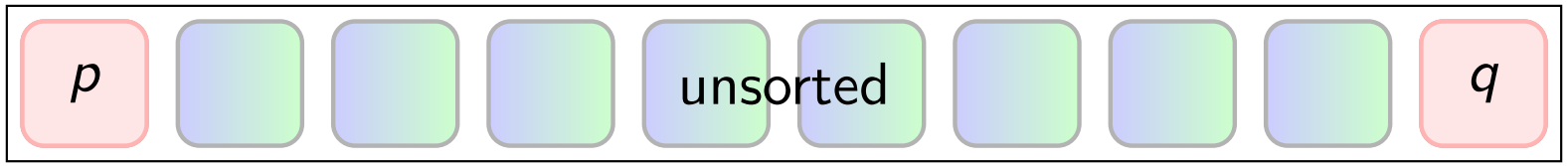
Dual Pivot Quicksort & Quickselect



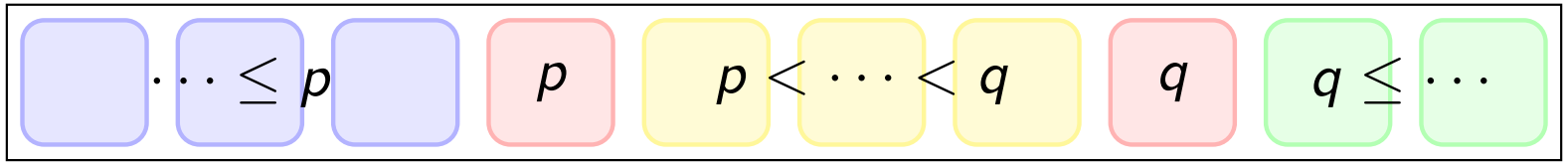
- choose pivot elements p and q
- partition into
  - small elements
  - medium elements
  - large elements



Dual Pivot Quicksort & Quickselect

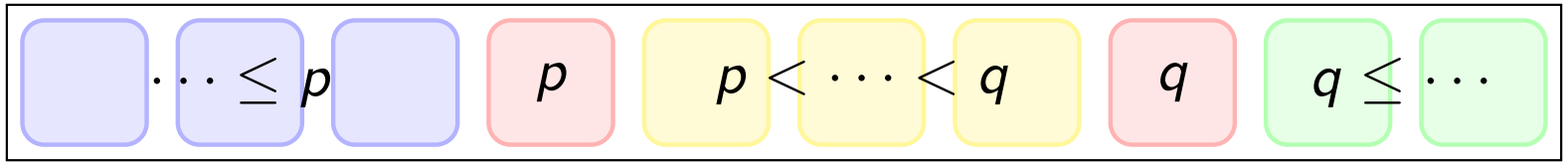


- choose pivot elements p and q
- partition into
  - small elements
  - medium elements
  - large elements



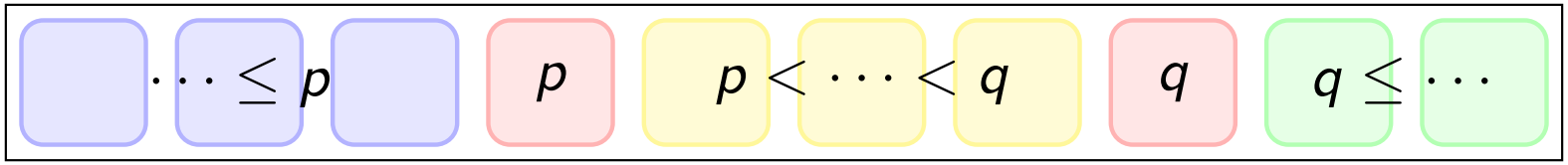
- proceed recursively

Average Number of Key Comparisons



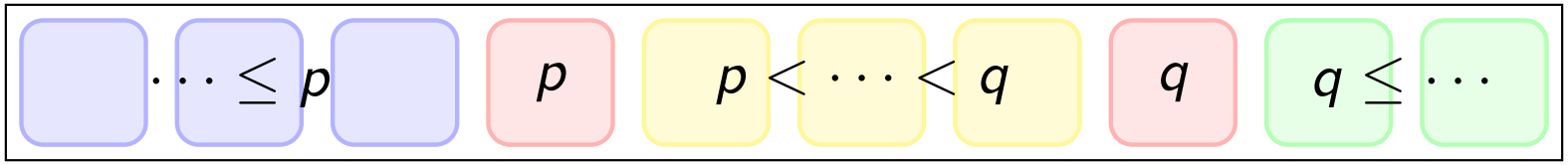
- partitioning
  - "classical" $\rightsquigarrow$ $n - 1$
- quicksort
  - "classical" $\rightsquigarrow$ $2n \log n - (2.84\ldots)n + O(\log n)$

Average Number of Key Comparisons



- partitioning
  - "classical" $\rightsquigarrow$ $n - 1$
- quicksort
  - "classical" $\rightsquigarrow$ $2n \log n - (2.84\ldots)n + O(\log n)$
  - "Yaroslavskiy–Bentley–Bloch" $\rightsquigarrow$ $1.9n \log n - (2.46\ldots)n + O(\log n)$
    [Wild–Nebel 2012]

Average Number of Key Comparisons



- partitioning
  - "classical" $\rightsquigarrow$ $n - 1$
  - "optimal dual pivot" $\rightsquigarrow$ $1.5n + 0.25 \log n + O(1)$
    [Aumüller–Dietzfelbinger 2014,
    Aumüller–Dietzfelbinger–Heuberger–K–Prodinger 2016]
- quicksort
  - "classical" $\rightsquigarrow$ $2n \log n - (2.84\ldots)n + O(\log n)$
  - "Yaroslavskiy–Bentley–Bloch" $\rightsquigarrow$ $1.9n \log n - (2.46\ldots)n + O(\log n)$
    [Wild–Nebel 2012]
  - "optimal dual pivot" $\rightsquigarrow$ $1.8n \log n - (2.38\ldots)n + O(\log n)$
    [Aumüller–Dietzfelbinger 2014,
    Aumüller–Dietzfelbinger–Heuberger–K–Prodinger 2016]

Overview

- C_n number of key comparisons using multi-pivot quicksort to sort a list with n elements

Questions

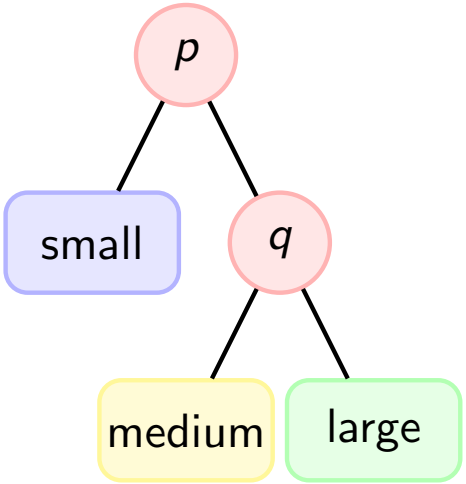
- What is the optimal multi-pivot strategy?
- Precise analysis of expected value of C_n?

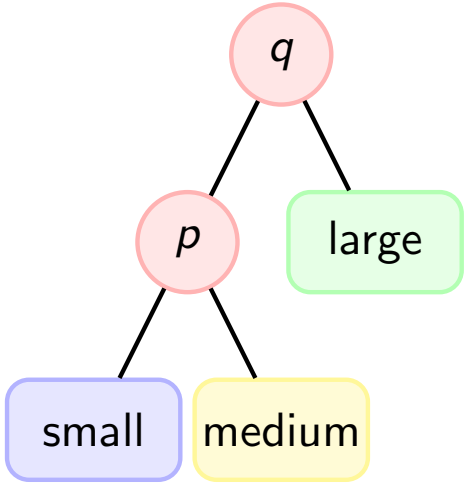
Optimal Partitioning Strategy "Count"

- comparison of element with pivots:
  - seen more smaller elements $\rightsquigarrow$ smaller pivot p first
  - seen more larger elements $\rightsquigarrow$ larger pivot q first
  - equality $\rightsquigarrow$ choice: smaller pivot p first

Optimal Partitioning Strategy "Count"

- comparison of element with pivots:
  - seen more smaller elements $\rightsquigarrow$ smaller pivot p first
  - seen more larger elements $\rightsquigarrow$ larger pivot q first
  - equality $\rightsquigarrow$ choice: smaller pivot p first
- comparison trees:





choose this tree if

#small $\geq$ #large

choose this tree if

#small $<$ #large

Probability Model

Input model

- random permutation of $\{1, \ldots, n\}$
- all $n!$ permutations equally likely

- random variables $S_0, \ldots, S_d$
  "counting classified elements"
  - e.g. $d = 2$ (dual-pivot)
    - $S_0 =$ #small
    - $S_1 =$ #medium
    - $S_2 =$ #large

Probability Model

Input model

- random permutation of $\{1, \ldots, n\}$
- all $n!$ permutations equally likely

- random variables $S_0, \ldots, S_d$
  "counting classified elements"
  - e.g. $d = 2$ (dual-pivot)
    - $S_0 = \#\text{small}$
    - $S_1 = \#\text{medium}$
    - $S_2 = \#\text{large}$
  - uniformly distributed
    with $S_0 + \cdots + S_d = n - d$
  - S_i identically distributed
    $\mathbb{P}(S_i = s_i) = \frac{1}{n^{\underline{d}}} d(n - s_i - 1)^{\underline{d-1}}$
- urn model

Probability Model

Input model

- random permutation of $\{1, \dots, n\}$
- all $n!$ permutations equally likely

- random variables $S_0, \dots, S_d$ "counting classified elements"
  - e.g. $d = 2$ (dual-pivot)
    - $S_0 =$ #small
    - $S_1 =$ #medium
    - $S_2 =$ #large
  - uniformly distributed with $S_0 + \cdots + S_d = n - d$
  - S_i identically distributed $\mathbb{P}(S_i = s_i) = \frac{1}{n^{\underline{d}}} d(n - s_i - 1)^{\underline{d-1}}$
- urn model

- random variables $S_0^{\leq k}, \dots, S_d^{\leq k}$ "already classified elements in step k"
  - uniformly distributed with $S_0^{\leq k} + \cdots + S_d^{\leq k} = k$
  - $\mathbb{P}\big(\text{next element is of type } i \mid (S_0^{\leq k}, \dots, S_d^{\leq k}) = (s_0, \dots, s_d)\big) = \dfrac{s_i + 1}{k + d + 1}$

Optimality

Theorem (ADH*K*P 2016, . . . , H*K* 2017)

- *cost* $P_n^\star$ *of strategy "Count"*
- *then* $\mathbb{E}(P_n^\star)$ *is minimal among all strategies*

Optimality

Theorem (ADH*K*P 2016, . . . , H*K* 2017)

- *cost $P_n^\star$ of strategy "Count"*
- *then $\mathbb{E}(P_n^\star)$ is minimal among all strategies*

- in step k
  - already classified elements
    e.g. $d = 2$ (dual-pivot)
    - $s_0 =$ #small
    - $s_1 =$ #medium
    - $s_2 =$ #large
  - classify next element
    - classification tree t
    - #comparisons
      $=$ depths $h_i(t)$ in tree t

Optimality

Theorem (ADH*K*P 2016, . . . , H*K* 2017)

- *cost $P_n^\star$ of strategy "Count"*
- *then $\mathbb{E}(P_n^\star)$ is minimal among all strategies*

- in step k
  - already classified elements e.g. $d = 2$ (dual-pivot)
    - $s_0 =$ #small
    - $s_1 =$ #medium
    - $s_2 =$ #large
  - classify next element
    - classification tree t
    - #comparisons $=$ depths $h_i(t)$ in tree t

- conditional cost
  - #comparisons to classify next element

$$\sum_{i=0}^{d} h_i(t) \underbrace{\frac{s_i + 1}{k + d + 1}}_{\substack{\text{conditional probability that} \\ \text{next element is of type } i}} = \frac{\ell_t(s)}{k + d + 1}$$

  - with affine linear function $\ell_t(s)$

Optimality

Theorem (ADH*K*P 2016, . . . , H*K* 2017)

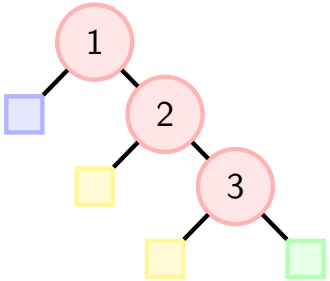
- *cost $P_n^\star$ of strategy "Count"*
- *then $\mathbb{E}(P_n^\star)$ is minimal among all strategies*

- in step k
  - already classified elements e.g. $d = 2$ (dual-pivot)
    - $s_0 =$ #small
    - $s_1 =$ #medium
    - $s_2 =$ #large
  - classify next element
    - classification tree t
    - #comparisons $=$ depths $h_i(t)$ in tree t
- conditional cost
  - #comparisons to classify next element

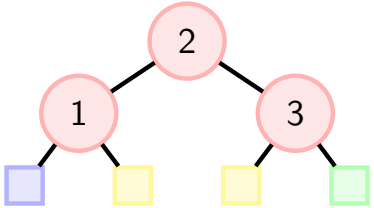
$$\sum_{i=0}^{d} h_i(t) \underbrace{\frac{s_i + 1}{k + d + 1}}_{\substack{\text{conditional probability that} \\ \text{next element is of type } i}} = \frac{\ell_t(s)}{k + d + 1}$$

  - with affine linear function $\ell_t(s)$
- de-condition (summing up)
- choose tree $T_k^\star = t$ such that $\ell_t(s)$ is minimal

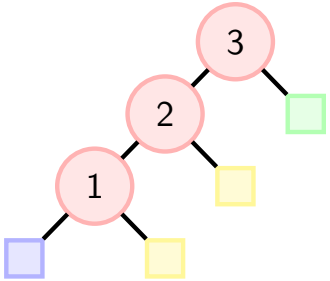
Optimal Partitioning Strategy with Three Pivots



$$s_1 \geq s_3$$
$$s_0 \geq s_2 + s_3 + 1$$

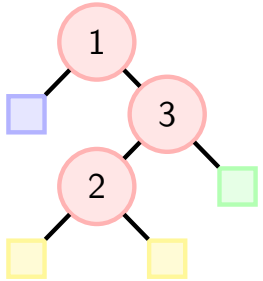


$$s_2 + s_3 + 1 \geq s_0$$
$$s_1 + s_2 + 1 \geq s_0$$
$$s_1 + s_2 + 1 \geq s_3$$
$$s_0 + s_1 + 1 \geq s_3$$

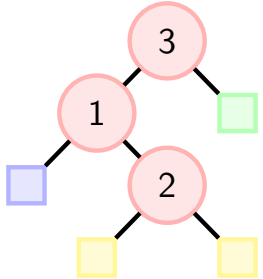


$$s_3 \geq s_0 + s_1 + 1$$
$$s_2 \geq s_0$$

$$s_0 \geq s_1 + s_2 + 1$$
$$s_3 \geq s_1$$
$$s_0 \geq s_3$$



$$s_0 \geq s_2$$
$$s_3 \geq s_0$$
$$s_3 \geq s_1 + s_2 + 1$$



comparison trees & polyhedra minimizing $\ell_t(s) = \sum_{i=0}^{d} h_i(t)(s_i + 1)$

Expected Partitioning Cost

- from optimality

$$\mathbb{E}(P_n) \geq \mathbb{E}(P_n^\star) = \mathbb{E}(Q_n) + \sum_{k=0}^{n-d-1} \frac{1}{(k+d+1)\binom{k+d}{d}} \underbrace{\sum_{s \in \mathcal{N}_k} \ell_{T_k^\star}(s)}_{= \sum_{t \in \mathcal{T}} \sum_{s \in \mathcal{N}_k} [s \in \mathcal{C}_t] \underbrace{\ell_t(s)}_{= \sum_{i=0}^{d} h_i(t)(s_i+1)}}$$

Expected Partitioning Cost

- from optimality

$$\mathbb{E}(P_n) \geq \mathbb{E}(P_n^\star) = \mathbb{E}(Q_n) + \sum_{k=0}^{n-d-1} \frac{1}{(k+d+1)\binom{k+d}{d}} \underbrace{\sum_{s\in\mathcal{N}_k} \ell_{T_k^\star}(s)}_{=\sum_{t\in\mathcal{T}}\sum_{s\in\mathcal{N}_k}[s\in\mathcal{C}_t]\underbrace{\ell_t(s)}_{=\sum_{i=0}^{d} h_i(t)(s_i+1)}}$$

- generating function $P(z) = \sum_{n\geq 0} \mathbb{E}(P_n)\, z^n$ satisfies

$$\left(\frac{\mathrm{d}}{\mathrm{d}z}\right)^{d+1}((1-z)\,P(z)) = d! \sum_{t\in\mathcal{T}} \sum_{i=0}^{d} h_i(t) \sum_{k\geq 0} \sum_{s\in\mathcal{N}_k} [s\in\mathcal{C}_t]\,(s_i+1)z^k = R(z)$$

Expected Partitioning Cost

- from optimality

$$\mathbb{E}(P_n) \geq \mathbb{E}(P_n^\star) = \mathbb{E}(Q_n) + \sum_{k=0}^{n-d-1} \frac{1}{(k+d+1)\binom{k+d}{d}} \underbrace{\sum_{s\in\mathcal{N}_k} \ell_{T_k^\star}(s)}_{=\sum_{t\in\mathcal{T}}\sum_{s\in\mathcal{N}_k}[s\in\mathcal{C}_t]\underbrace{\ell_t(s)}_{=\sum_{i=0}^d h_i(t)(s_i+1)}}$$

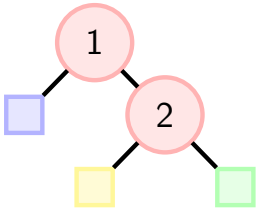
- generating function $P(z) = \sum_{n\geq 0} \mathbb{E}(P_n)\, z^n$ satisfies

$$\left(\frac{\mathrm{d}}{\mathrm{d}z}\right)^{d+1}((1-z)\,P(z)) = d! \sum_{t\in\mathcal{T}} \sum_{i=0}^{d} h_i(t) \underbrace{\sum_{k\geq 0} \sum_{s\in\mathcal{N}_k} [s\in\mathcal{C}_t]\,(s_i+1)z^k}_{=\sum_{s\in\mathcal{C}_t}(s_i+1)z^{s_0+\cdots+s_d} \;=\; \frac{\partial}{\partial y} G_{t,i}(y,z)\big|_{y=1}} = R(z)$$

- $G_{t,i}(y,z) = y\, H_t(z,\ldots,z,zy,z,\ldots,z)$
  with $H_t(y_0,\ldots,y_d) = \sum_{s\in\mathcal{C}_t} y^s$ and polyhedron $\mathcal{C}_t$

How to Compute Generating Function of Polyhedron?

- $d = 2$, $\mathcal{C}_{t_1} = \{(s_0, s_1, s_2) \in \mathbb{N}_0^4 \mid s_0 \geq s_2\}$

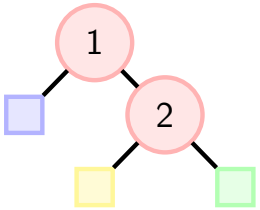


$$H_{t_1}(y_0, y_1, y_2) = \sum_{s_0 \geq s_2 \geq 0, s_1 \geq 0} y_0^{s_0} y_1^{s_1} y_2^{s_2} = \sum_{s_1, s_2, u \geq 0} y_0^{s_2+u} y_1^{s_1} y_2^{s_2}$$

$$= \sum_{s_1, s_2, u \geq 0} y_0^{u} y_1^{s_1} (y_0 y_2)^{s_2} = \frac{1}{(1-y_0)(1-y_1)(1-y_0 y_2)}$$

How to Compute Generating Function of Polyhedron?

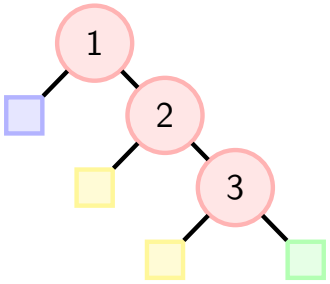
- $d = 2$, $\mathcal{C}_{t_1} = \{(s_0, s_1, s_2) \in \mathbb{N}_0^4 \mid s_0 \geq s_2\}$



$$H_{t_1}(y_0, y_1, y_2) = \sum_{s_0 \geq s_2 \geq 0, s_1 \geq 0} y_0^{s_0} y_1^{s_1} y_2^{s_2} = \sum_{s_1, s_2, u \geq 0} y_0^{s_2+u} y_1^{s_1} y_2^{s_2}$$

$$= \sum_{s_1, s_2, u \geq 0} y_0^{u} y_1^{s_1} (y_0 y_2)^{s_2} = \frac{1}{(1-y_0)(1-y_1)(1-y_0 y_2)}$$

- $d = 3$, $\mathcal{C}_{t_1} = \{(s_0, s_1, s_2, s_3) \in \mathbb{N}_0^4 \mid s_1 \geq s_3,\ s_0 \geq s_2 + s_3 + 1\}$



$$H_{t_1}(y_0, y_1, y_2, y_3) = \sum_{s_1 \geq s_3 \geq 0, s_0 \geq s_2 + s_3 + 1, s_2 \geq 0} y_0^{s_0} y_1^{s_1} y_2^{s_2} y_3^{s_3}$$

$$= \sum_{s_2, s_3, u, v \geq 0} y_0^{s_2+s_3+v+1} y_1^{s_3+u} y_2^{s_2} y_3^{s_3}$$

$$= \frac{y_0}{1-y_0} \frac{1}{1-y_1} \frac{1}{1-y_0 y_2} \frac{1}{1-y_0 y_1 y_3}$$

MacMahon's Omega Calculus

Ω-Operator

$$\underset{\geq}{\Omega} \sum_{s \in \mathbb{N}_0^{d+1}} \sum_{r \in \mathbb{Z}^m} c_{sr} y^s \lambda^r = \sum_{s \in \mathbb{N}_0^{d+1}} \sum_{r \in \mathbb{N}_0^m} c_{sr} y^s$$

- applying
  - remove summands corresponding to negative exponents of λ
  - keep summands corresponding to non-negative exponents of λ

MacMahon's Omega Calculus

Ω-Operator

$$\underset{\geq}{\Omega} \sum_{s \in \mathbb{N}_0^{d+1}} \sum_{r \in \mathbb{Z}^m} c_{sr} y^s \lambda^r = \sum_{s \in \mathbb{N}_0^{d+1}} \sum_{r \in \mathbb{N}_0^m} c_{sr} y^s$$

- applying
  - remove summands corresponding to negative exponents of λ
  - keep summands corresponding to non-negative exponents of λ

- polyhedron $\mathcal{C} = \{s \in \mathbb{N}_0^{d+1} \mid Ms \geq -b\}$
- inequality contributes $\lambda_j^{\sum_{i=0}^d M_{ji} s_i + b_j}$
- generating function

$$H(y_0, \ldots, y_d) = \sum_{s \in \mathcal{C}} y^s = \underset{\geq}{\Omega} \frac{\prod_j \lambda_j^{b_j}}{\prod_{i=0}^d \left(1 - y_i \prod_j \lambda_j^{M_{ji}}\right)}$$

Applying Omega

- based on partial fraction decomposition
- hundreds of thousands summands in multivariate Laurent polynomials
- cut polyhedron into many parts
- handle equations (and moduli)
- handle "simple" inequalities directly

Applying Omega

- based on partial fraction decomposition
- hundreds of thousands summands in multivariate Laurent polynomials
- cut polyhedron into many parts
- handle equations (and moduli)
- handle "simple" inequalities directly

Result for $d = 4$

$$(1-z)^4\, R(z) = \frac{19072}{75(1-z)^2} + \frac{1744}{75(1-z)} - \frac{48(3z^2-z+3)}{5(1+z+z^2+z^3+z^4)^2} + \frac{48(51z^3+14z^2+14z+51)}{25(1+z+z^2+z^3+z^4)} + \frac{24}{(1+z+z^2)^3} + \frac{8(3z-2)}{(1+z+z^2)^2} - \frac{8(19z+16)}{3(1+z+z^2)} - \frac{24}{1+z}$$

with $R(z) = \left(\frac{\mathrm{d}}{\mathrm{d}z}\right)^{d+1}((1-z)\,P(z))$

Solving the Multi-Pivot Quicksort Recurrence

- expected values

$$C_n = P_n^\star + \sum_{i=0}^{d} C_{S_i} \quad \Longrightarrow \quad \mathbb{E}(C_n) = \mathbb{E}(P_n^\star) + \sum_{i=0}^{d} \sum_{s_i=0}^{n-d} \mathbb{E}(C_{s_i})\, \mathbb{P}(S_i = s_i)$$

Solving the Multi-Pivot Quicksort Recurrence

- expected values

$$C_n = P_n^\star + \sum_{i=0}^{d} C_{S_i} \quad \Longrightarrow \quad \mathbb{E}(C_n) = \mathbb{E}(P_n^\star) + \sum_{i=0}^{d} \sum_{s_i=0}^{n-d} \mathbb{E}(C_{s_i})\, \mathbb{P}(S_i = s_i)$$

- differential equation

$$(1-z)^d \left(\frac{\mathrm{d}}{\mathrm{d}z}\right)^d C(z) - (d+1)!\, C(z) = (1-z)^d \left(\frac{\mathrm{d}}{\mathrm{d}z}\right)^d P(z)$$

Solving the Multi-Pivot Quicksort Recurrence

- expected values

$$C_n = P_n^\star + \sum_{i=0}^{d} C_{S_i} \quad \Longrightarrow \quad \mathbb{E}(C_n) = \mathbb{E}(P_n^\star) + \sum_{i=0}^{d} \sum_{s_i=0}^{n-d} \mathbb{E}(C_{s_i})\, \mathbb{P}(S_i = s_i)$$

- differential equation

$$(1-z)^d \left(\frac{\mathrm{d}}{\mathrm{d}z}\right)^d C(z) - (d+1)!\, C(z) = (1-z)^d \left(\frac{\mathrm{d}}{\mathrm{d}z}\right)^d P(z)$$

- solution

$$C(z) = Q(z) + (I_{\alpha_1} \circ I_{\alpha_2} \circ \cdots \circ I_{\alpha_d}) \left((d+1)!\, Q(z) + (1-z)^d \left(\frac{\mathrm{d}}{\mathrm{d}z}\right)^d P(z) \right)$$

with

- $(I_\alpha f)(z) = (1-z)^{-\alpha} \int_0^z (1-t)^{\alpha-1} f(t)\, dt$
- polynomial $Q(z)$
- $\sum_{k=1}^{d} \left[{d \atop k}\right] X^k - (d+1)! = \prod_{i=1}^{d} (X - \alpha_i)$

Singularity Analysis & Transfers

- operator I_α for $z \to 1$
  - $I_\alpha \dfrac{1}{(1-z)^\beta} = \dfrac{1}{\alpha-\beta}\left(\dfrac{1}{(1-z)^\alpha} - \dfrac{1}{(1-z)^\beta}\right)$

    for $\alpha \neq \beta$
  - $(I_{\alpha_1} \circ \cdots \circ I_{\alpha_k})\,(1-z)^{-\beta}\big(-\log(1-z)\big)^\ell = \ldots$

Singularity Analysis & Transfers

- operator I_α for $z \to 1$
  - $I_\alpha \dfrac{1}{(1-z)^\beta} = \dfrac{1}{\alpha-\beta}\left(\dfrac{1}{(1-z)^\alpha} - \dfrac{1}{(1-z)^\beta}\right)$

    for $\alpha \neq \beta$
  - $(I_{\alpha_1} \circ \cdots \circ I_{\alpha_k})\,(1-z)^{-\beta}\big(-\log(1-z)\big)^\ell = \ldots$

Transfer

- $f(z) = O\big((1-z)^{-\beta}\big)$ for $z \to 1$

  with $f^\star(s) = \int_0^1 (1-t)^{s-1} f(t)\,\mathrm{d}t$

$$(I_\alpha f)(z) = \frac{f^\star(\alpha)}{(1-z)^\alpha} + O\big((1-z)^{-\beta}\big)$$

for $\operatorname{Re}\beta < \operatorname{Re}\alpha$

Singularity Analysis & Transfers

- operator I_α for $z \to 1$
  - $I_\alpha \dfrac{1}{(1-z)^\beta} = \dfrac{1}{\alpha-\beta}\left(\dfrac{1}{(1-z)^\alpha} - \dfrac{1}{(1-z)^\beta}\right)$

    for $\alpha \neq \beta$
  - $(I_{\alpha_1} \circ \cdots \circ I_{\alpha_k})\,(1-z)^{-\beta}\big(-\log(1-z)\big)^\ell = \ldots$

Transfer

- $f(z) = O\big((1-z)^{-\beta}\big)$ for $z \to 1$

  with $f^\star(s) = \int_0^1 (1-t)^{s-1} f(t)\,\mathrm{d}t$

$$(I_\alpha f)(z) = \frac{f^\star(\alpha)}{(1-z)^\alpha} + O\big((1-z)^{-\beta}\big)$$

  for $\operatorname{Re}\beta < \operatorname{Re}\alpha$

- operator I_α for $z \to \rho \neq 1$, e.g. $d = 4$, $\rho = \zeta_{60}^k$
  - $(I_{\alpha_1} \circ \cdots \circ I_{\alpha_k})\,\big(1-\frac{z}{\rho}\big)^{-\beta}\big(-\log\big(1-\frac{z}{\rho}\big)\big)^\ell = \ldots$

The Result

Theorem (ADH*K*P 2016)

expected number of key comparisons in dual pivot quicksort with the **optimal** *partitioning strategy "Count" is*

$$\frac{9}{5} n H_n - \frac{1}{5} n H_n^{\mathrm{alt}} - \frac{89}{25} n + \frac{67}{40} H_n - \frac{3}{40} H_n^{\mathrm{alt}} - \frac{83}{800} + \frac{(-1)^n}{10} + \cdots$$

$$= \frac{9}{5} n \log n + An + B \log n + C + \frac{D}{n} + \frac{E}{n^2} + \frac{(-1)^n F + G}{n^3} + O\left(\frac{1}{n^4}\right)$$

asymptotically as $n \to \infty$

- *harmonic numbers*
  - $H_n = \sum_{i=1}^{n} 1/i$
  - $H_n^{\mathrm{alt}} = \sum_{i=1}^{n} (-1)^i / i$
- *constant of linear term*
  $A = \frac{9}{5}\gamma + \frac{1}{5}\log 2 - \frac{89}{25}$
  $= -2.3823823670652\ldots$
- *explicit constants* B, C, ...

The Result

Theorem (Heuberger–K 2017)

- *expected number of key comparisons*
  in trial pivot quicksort
  with the **optimal** *partitioning strategy is*

$$\frac{133}{78} n \log n + An + B \log n + O(1)$$

  - $A = \frac{133}{78}\gamma - \frac{2}{117}\sqrt{3}\pi + \frac{4}{39}\log 3 + \frac{3}{26}\log 2 - \frac{6761}{2028} = -2.24995\ldots$
  - $B = \frac{707}{468}$
- *expected number of key comparisons*
  in quadral pivot quicksort
  with the **optimal** *partitioning strategy is*

$$\frac{9536}{5775} n \log n + An + B \log n + O(1)$$

  - $A = \mathit{skipped} = -2.20515\ldots$
  - $B = \frac{48823}{34650}$