TÖL403G GREINING REIKNIRITA

3. Deila-og-Drottna 1

Hjálmtýr Hafsteinsson
Vor 2022

Í þessum fyrirlestri

- Deila-og-Drottna reiknirit
  - Almennt skipulag

- Rakningartré (*recursion trees*)
  - Aðferð til að leysa rakningarvensl

1.6 − 1.8

- Finna miðgildi á O(*n*) tíma (*selection*)
  - *QuickSelect*
  - Nota sniðuga aðferð til að velja vendistak

Deila-og-Drottna (*Divide-and-Conquer*)

- Sáum síðast nokkur reiknirit:
  - Turnarnir í Hanoi, Mergesort, Quicksort
- Þau höfðu öll svipað skipulag:

  - Skipta verkefninu upp í sjálfstæð undirverkefni
  - Leysa undirverkefnin endurkvæmt
  - Sameina lausnir undirverkefnanna

Fara **niður** endurkvæmnina

Fara **upp** endurkvæmnina

- Þegar undirverkefnin eru nógu lítil eru þau leyst beint
- Tímaflækjan er nær alltaf rakningarvensl (*recurrence*), sem þarf að leysa

Dæmi um rakningarvensl

- Turnarnir í Hanoi

$$T(n) = 2T(n-1) + 1$$
$$T(0) = 0$$

Tvær færslur á n−1 stafla

Færsla á neðstu skífu

Leystum með ágiskun og þrepunarsönnun

Lausn: $T(n) = 2^n - 1$

- Mergesort

$$T(n) = 2T(n/2) + n$$
$$T(1) = 1$$

Raða báðum helmingum

Samruni (*merge*)

Leystum með ágiskun og þrepunarsönnun

Lausn: $T(n) = O(n \log n)$

Rakningartré

- Fáum oft rakningarvensl af gerðinni:

$$T(n) = rT(n/c) + f(n)$$

r : fjöldi uppskiptinga (2 í Mergsort)
n/c : stærð hvers hlutverkefnis
$f(n)$: kostn. við uppskiptingu og sameiningu lausna

- Aðferð: Setjum endurkvæmnina upp sem tré og skráum kostnaðinn við hvert undirverkefni
- Summa kostnaðar yfir allt tréð er þá heildartími reikniritsins
  - Þægilegast að finna fyrst summu hvers lags (*level*) trésins og leggja þau gildi svo saman

Smíði rakningartrés fyrir $T(n) = rT(n/c) + f(n)$

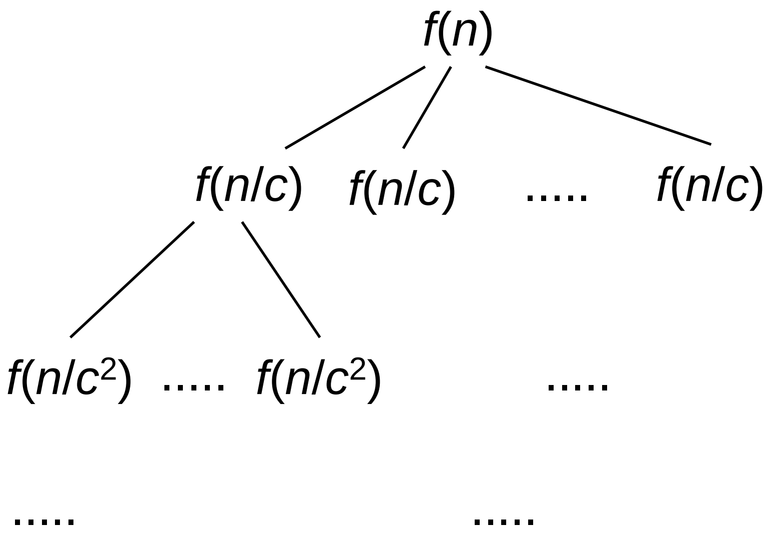
Tréð hefur $f(n)$ kostnað í rótinni (efsta lag)

síðan eru r börn, hvert þeirra með $f(n/c)$ kostnað

börn þeirra hafa kostnaðinn $f(n/c^2)$, o.s.frv.

Dýpi trésins er $L = \log_c n$

því við hættum þegar $n/c^L = 1$



<u>Summa hvers lags</u>

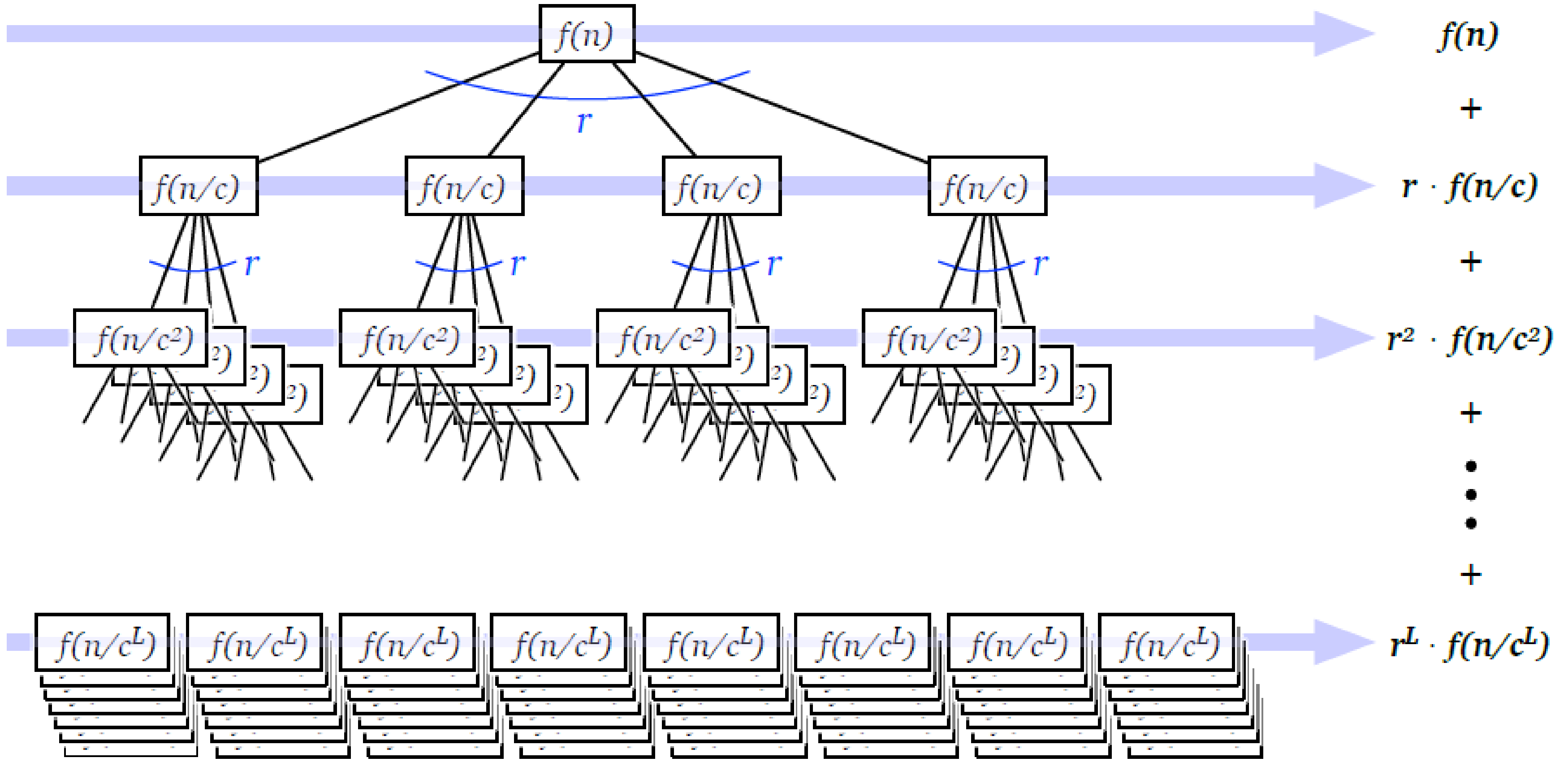
$f(n)$

$r \cdot f(n/c)$

$r^2 \cdot f(n/c^2)$

$r^k \cdot f(n/c^k)$ Almennt, á lagi k

Rakningartré



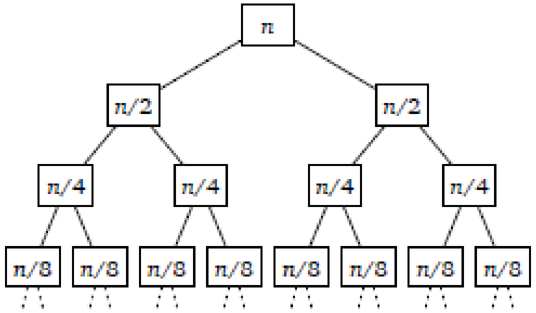
Fer eftir *r*, *c* og *f*(*n*) hversu auðvelt er að einfalda þetta

Leggjum saman summur fyrir hvert lag: $T(n) = \sum_{i=0}^{L} r^i f(\frac{n}{c^i})$

Notkun rakningatrjáa

- <u>Mergsort</u>: $T(n) = 2T(n/2) + n$
  $T(1) = 1$

Hér er $r = 2$, $c = 2$ og $f(n) = n$

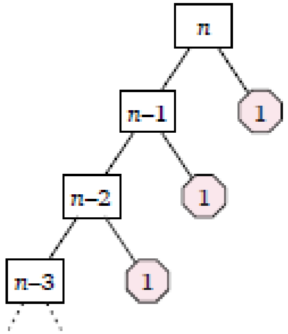


Þá er summan:

$$T(n) = \sum_{i=0}^{\log n} 2^i \cdot \frac{n}{2^i} = \sum_{i=0}^{\log n} n = \boldsymbol{n \log n}$$

- <u>Quicksort</u>: $T(n) = T(n-1) + T(1) + O(n)$

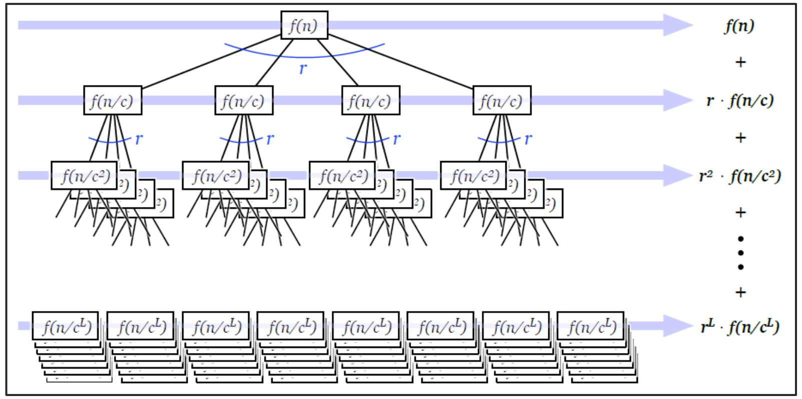
Þetta passar ekki inn í formúluna, en við getum notað rakningartré til að finna lausnina



Hér er summa laganna: $n + n + (n\text{-}1) + (n\text{-}2) + \ldots + 1$ Augljóslega $\boldsymbol{O(n^2)}$

Algeng tilvik

- Þegar við leysum summu laganna koma upp 3 tilvik eftir því hvernig liðir summunnar hegða sér:



<u>Lækkandi</u>:

Summan er lækkandi kvótaröð (hver liður er fasta lægri en síðasti liður)

Lausn: $T(n) = O(f(n))$

Rótin ræður!

<u>Jafnir</u>:

Allir liðirnir jafnir, svo við margföldum með fjölda þeirra

Lausn: $T(n) = O(f(n) \cdot L) = O(f(n) \cdot \log n)$

Margfaldast með $O(\log(n))$!

<u>Hækkandi</u>:

Summan er hækkandi kvótaröð (hver liður er fasta hærri en síðasti liður)

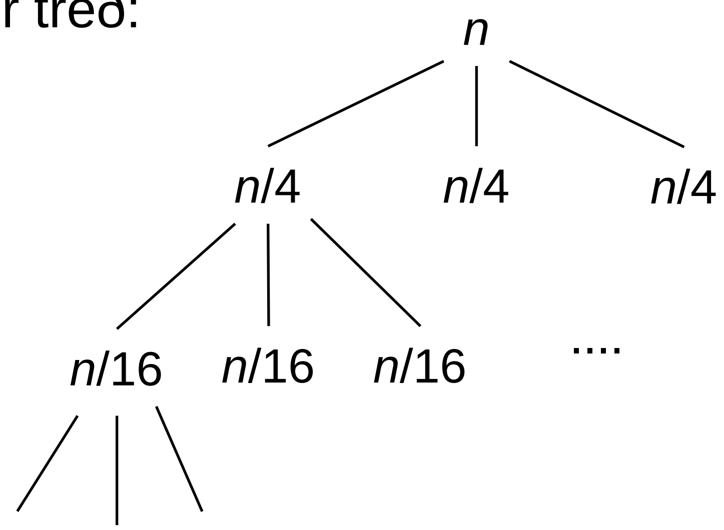
Lausn: $T(n) = O\left(n^{\log_c r} \cdot T(1)\right) = O(n^{\log_c r})$

Laufin ráða!

Dæmi um tilvik

- $T(n) = 3T(n/4) + n$

Þá er tréð:



<u>Summa laga</u>

n

$3*n/4 = (3/4)n$

$9*n/16 = (3/4)^2 n$

⋮

$$T(n) = \sum_{i=0}^{L} (3/4)^i n$$

Athugið að það er oft gott að einfalda segðirnar í hverju lagi til að finna einhverja reglu

Þetta er <u>lækkandi</u> kvótaröð svo lausnin er $O(n)$

Dæmi um rakningarvensl

- $T(n) = T(n/3) + n^2$

Þá er summan: $\sum_{i=0}^{L} 1^i (\frac{n}{3^i})^2 = \sum_{i=0}^{L} \frac{1}{9^i} n^2$ Lækkandi kvótaröð!

Lausn: $T(n) = O(n^2)$

- $T(n) = 2T(n/2) + n$

Þá er summan: $\sum_{i=0}^{L} 2^i \frac{n}{2^i} = \sum_{i=0}^{L} n$ Jafnir liðir!

Lausn: $T(n) = O(n \log n)$

- $T(n) = 3T(n/2) + n$

Þá er summan: $\sum_{i=0}^{L} 3^i \frac{n}{2^i} = \sum_{i=0}^{L} (3/2)^i n$ Hækkandi kvótaröð!

Lausn: $T(n) = O\left(n^{\log_2 3}\right) = O(n^{1.5849\ldots})$

Finna miðgildi (*median*)

- Verkefni: Finna k-ta minnsta stakið í n-staka óröðuðum lista
- Hugmynd: Nota sömu hugmynd og Quicksort: skipta listanum eftir vendistaki og leita svo áfram í þeim hluta þar sem stakið hlýtur að vera

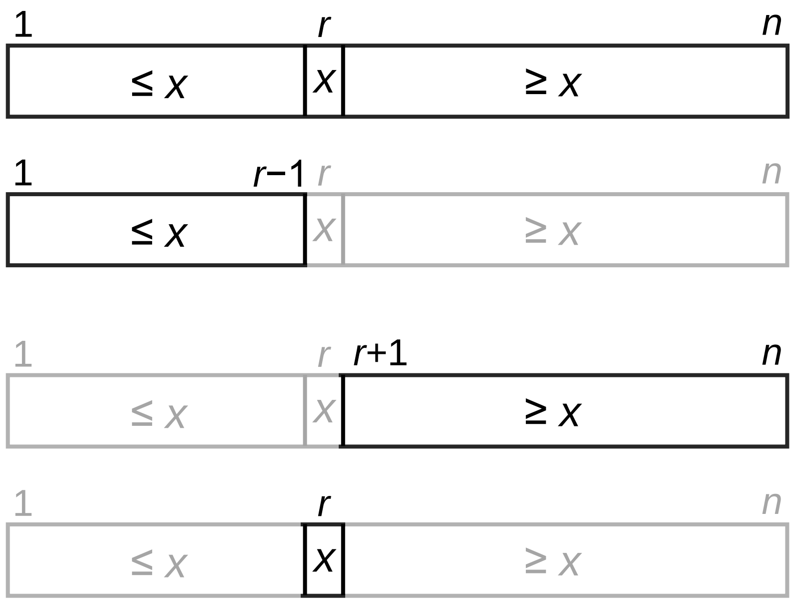
Notum $A[p]$ ($=x$) sem vendistak í listanum $A[1..n]$

Skiptum $A[1..n]$ um x

Ef $k < r$, þá eigum við að leita að k-ta minnsta stakinu í vinstri hlutanum, þ.e. $A[1..r-1]$

Ef $k > r$, þá eigum við að leita að $(k-r)$-ta minnsta stakinu í hægri hlutanum, þ.e. $A[r+1..n]$

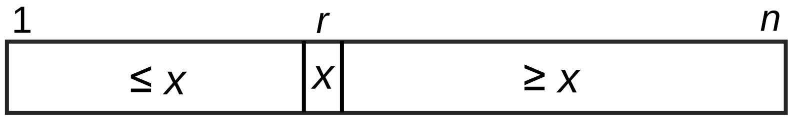
Ef $k = r$, þá er k-ta minnsta stakið fundið!



Reiknritið *QuickSelect*

```
QuickSelect(A[1..n], k):
  if n = 1
      return A[1]
  else
      Choose a pivot element A[p]
      r ← Partition(A[1..n], p)

      if k < r
          return QuickSelect(A[1..r − 1], k)
      else if k > r
          return QuickSelect(A[r + 1..n], k − r)
      else
          return A[r]
```



Upphaflega sett fram af Tony Hoare 1961 í sömu grein og Quicksort!

Tímaflækjan ræðst af vali á vendistakinu:

Ef við erum alltaf óheppin þá eru rakningarvenslin $T(n) = T(n-1) + O(n)$ sem þýðir að $T(n) = O(n^2)$

Árið 1972 fundu Blum, Floyd, Pratt, Rivest og Tarjan leið til að velja vendistakið þannig að tímaflækjan verður aðeins $O(n)$

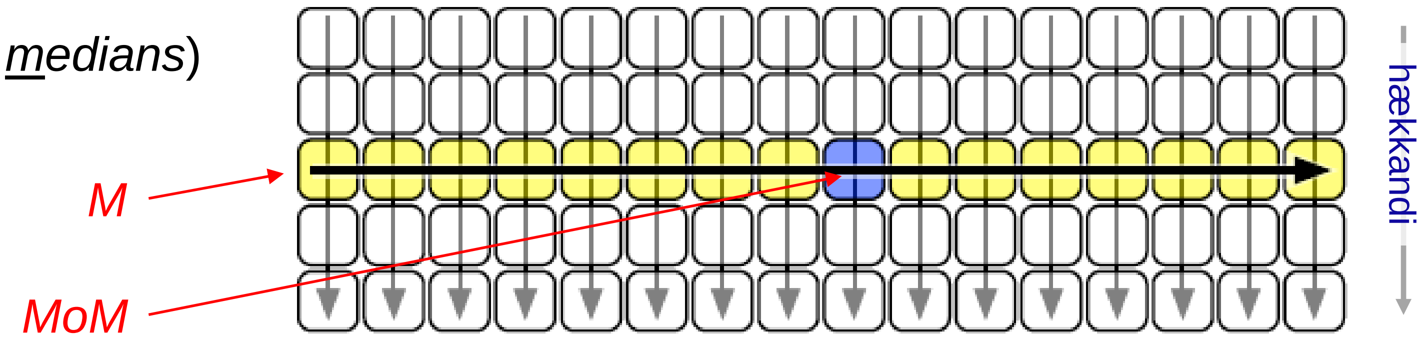
Blum-Floyd-Pratt-Rivest-Tarjan

- Aðferð þeirra eyðir nokkru púðri í að finna gott vendistak, sem tryggir að vendistakið sé aldrei mjög langt frá miðgildinu
- Aðferðin:
  - Skipta fylkinu A[1..*n*] upp í $\lceil n/5 \rceil$ blokkir, sem hver hefur 5 stök
  - Finna miðgildin í hverri 5-staka blokk fyrir sig
  - Búum til fylkið M[1.. $\lceil n/5 \rceil$] og finnum miðgildi þess <u>endurkvæmt</u>!

Ef *n* er ekki deilanlegt með 5 þá er hægt að bæta við nokkrum ∞ stökum aftast

Hægt að finna miðgildi 5 staka með 6 samanburðum

Miðgildi fylkisins *M* kallast *MoM* (*<u>m</u>edian <u>o</u>f <u>m</u>edians*) er svo notað sem vendistak fyrir allt fylkið



Reikniritið *MoMSelect*

Notum nú nýja aðferð til að velja vendistakið (rautt):

$\underline{\textsc{MomSelect}(A[1..n], k):}$
  if $n \leq 25$ ⟨⟨*or whatever*⟩⟩
    use brute force
  else
    $m \leftarrow \lceil n/5 \rceil$
    for $i \leftarrow 1$ to m
      $M[i] \leftarrow \textsc{MedianOfFive}(A[5i-4..5i])$ ⟨⟨*Brute force!*⟩⟩
    $mom \leftarrow \textsc{MomSelect}(M[1..m], \lfloor m/2 \rfloor)$ ⟨⟨*Recursion!*⟩⟩
    $r \leftarrow \textsc{Partition}(A[1..n], mom)$
    if $k < r$
      return $\textsc{MomSelect}(A[1..r-1], k)$ ⟨⟨*Recursion!*⟩⟩
    else if $k > r$
      return $\textsc{MomSelect}(A[r+1..n], k-r)$ ⟨⟨*Recursion!*⟩⟩
    else
      return mom

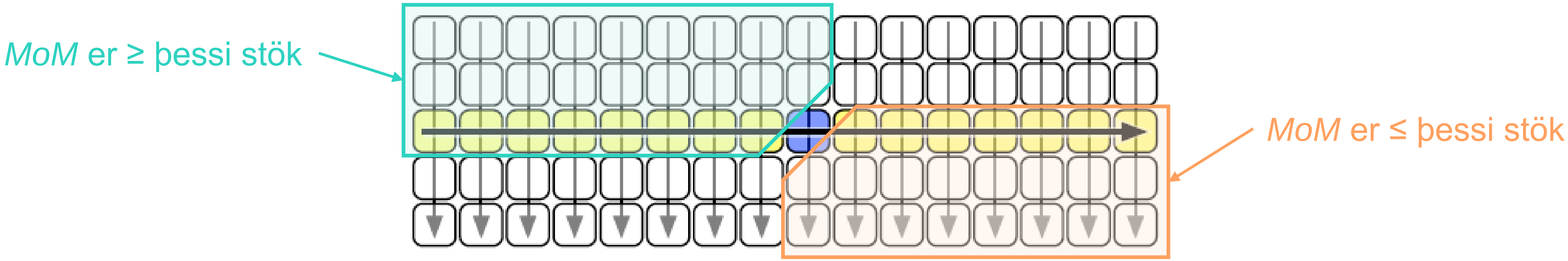
Viljum ekki nota endurkvæmu aðferðina fyrir lítil tilvik

Finna miðgildi 5 staka með 6 samanburðum

Finna svo miðgildi miðgildanna (MoM) endurkvæmt

Hversu gott er *MoM*?

- Miðgildi miðgildanna (*MoM*) getur aldrei orðið eitt af minnstu eða eitt af stærstu stökunum



MoM er örugglega ≥ einn hópur staka og líka örugglega ≤ annar jafnstór hópur staka

Hversu stórir eru þessir hópar?

Hvor hópur um sig inniheldur 3*n*/10 stök

Við náum að útiloka a.m.k. annan þeirra í endurkvæmninni

Þetta þýðir að versta mögulega skipting á stökunum er að það séu ***7n/10*** stök í þeim hluta sem *k*-ta minnsta stakið er í

Tímaflækja á *MoMSelect*

- Fáum rakningarvenslin

$$T(n) \leq T\left({}^{n}/_{5}\right) + T\left({}^{7n}/_{10}\right) + O(n)$$

Finna *MoM* í
M fylkinu

Endurkvæmni
í *A* fylkinu

Skipting
(*partition*)

Af hverju veljum við 5 stök í hóp?

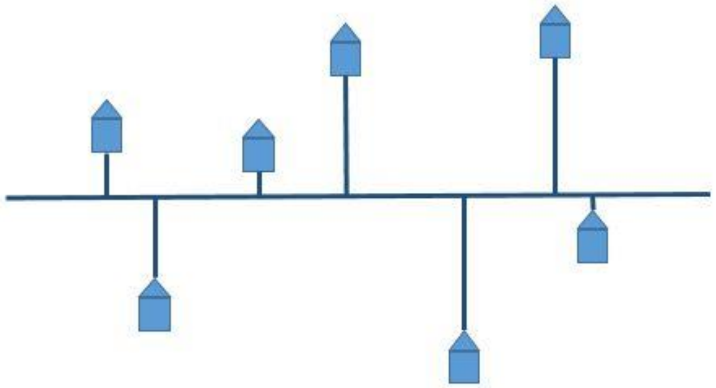
Athugið að $n/5 + 7n/10 = 9n/10$

Samanlögð stærð hlutverkefnanna minnkar eftir því
sem við förum dýpra í endurkvæmnina (n, $9n/10$, $81n/100$, ...)

Skoðum þetta betur næst!

Fyrirlestraæfingar

1. Leggja á leiðslu sem liggur beint í vestur-austur (þ.e. samsíða *x*-ás). Við viljum tengja *n* hús við leiðsluna, en heimtaugarnar liggja allar í norður-suður (þ.e. samsíða *y*-ás). Lágmarka á heildarlengd heimtauganna. Útskýrið hvernig við getum leyst þetta verkefni á $O(n)$ tíma ef við höfum *x*- og *y*-hnit allra húsanna.



2. Hver haldið þið að sé væntur tími (*expected time*) á *QuickSelect* ef við veljum vendistakið af handahófi (*random pivot*)? Rökstyðjið óformlega.

3. Hver væri mismunajafnan fyrir **MomSelect** ef hópastærðin væri 3? En ef hópastærðin væri 7?