TÖL403G GREINING REIKNIRITA

# 20. Jafnaðargreining 4

Hjálmtýr Hafsteinsson
Vor 2022

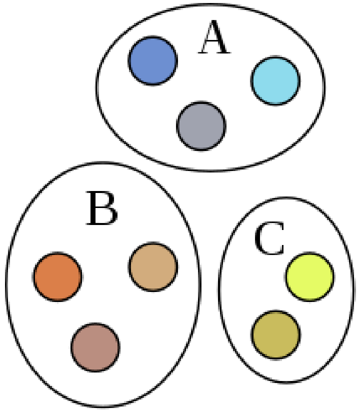
# Í þessum fyrirlestri

- Gagnagrind fyrir aðskilin mengi
  - *Union-Find* verkefnið
  - Sameining eftir þyngd
  - Vegþjöppun
  - Jafnaðargreining

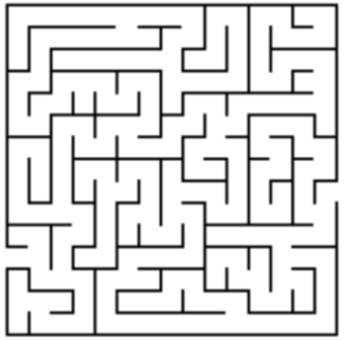
DC 11.1 − 11.3, 11.6 − 11.7

# Aðskilin mengi (*disjoint sets*)

- Viljum stundum skipta stökum upp í hópa
  - Hvert stak í nákvæmlega einum hópi
- Hóparnir eru þá aðskilin mengi




- Notkunardæmi;
  - Halda utanum samhangandi hluta nets (*connected components*)
  - Kruskal reikniritið fyrir léttasta spantré nets
  - Búa til völundarhús (*maze*)
  - Notað á ýmsum stöðum í þýðendum



# Auðkenning mengja

- Aðskildu mengin geta verið mismunandi mörg
  - Einnig getur fjöldi þeirra breyst
- Hvernig búum við til nöfn á mengin?

- Einföld leið:

Notum eitt af stökunum í menginu sem auðkenni

Köllum það aðalstak (*leader*) mengisins

Skiptir ekki máli hvaða stak við veljum, bara að allir séu sammála um hvað stak sé aðalstak mengisins

Sama stakið er alltaf aðalstak fyrir sama mengið

Þegar við sameinum tvö mengi þá veljum við annað aðalstakið sem nýtt aðalstak fyrir nýja mengið

# Aðgerðir á aðskilin mengi

- Skilgreinum þrjár aðgerðir á aðskilin mengi:

| | | |
|---|---|---|
| *MakeSet*(*x*) | • *GeraMengi*(*x*): | Býr til mengið {*x*}, með *x* sem aðalstak |
| *Find*(*x*) | • *Finna*(*x*): | Skila aðalstaki mengisins sem *x* er í |
| *Union*(*A*, *B*) | • *Sameina*(*A*, *B*): | Sameina mengin *A* og *B* ($A \cup B$) |

Í hvaða mengi er *x*?

Megum líka segja *Sameina*(*x*, *y*) í stað *Sameina*(*Finna*(*x*), *Finna*(*y*))

Þetta verkefni er oft kallað "*Union-Find*" verkefnið

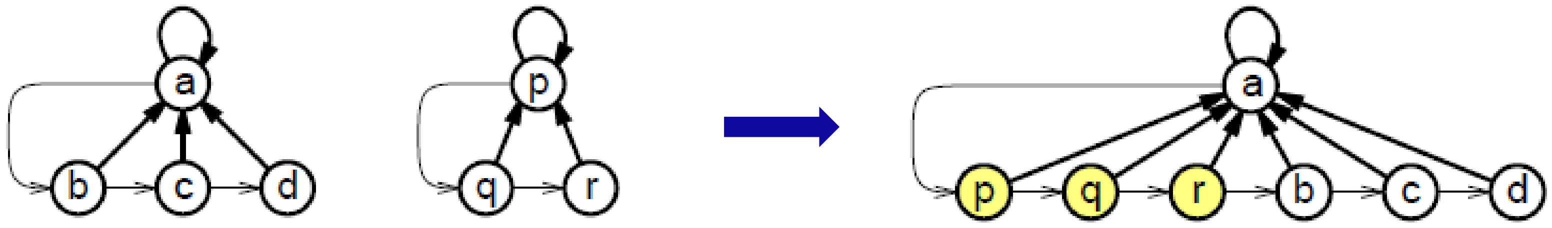
Þær tvær aðgerðir eru mikilvægastar og taka mestan tíma

# Útfærslur á *Union-Find*

- Tengdur listi með hausbendi

Haushnúturinn inniheldur aðalstak mengisins

Hvert mengi er tengdur listi og hver hnútur í honum hefur bendi á haushnútinn




$\textsc{Union}(x, y)$:
  $\bar{x} \leftarrow \textsc{Find}(x)$
  $\bar{y} \leftarrow \textsc{Find}(y)$

  $y \leftarrow \bar{y}$
  $\text{leader}(y) \leftarrow \bar{x}$
  while $(\text{next}(y) \neq \textsc{Null})$
    $y \leftarrow \text{next}(y)$
    $\text{leader}(y) \leftarrow \bar{x}$

  $\text{next}(y) \leftarrow \text{next}(\bar{x})$
  $\text{next}(\bar{x}) \leftarrow \bar{y}$

*GeraMengi*(*x*) og *Finna*(*x*) taka bæði *O*(1) tíma:

$\textsc{MakeSet}(x)$:
  $\text{leader}(x) \leftarrow x$
  $\text{next}(x) \leftarrow x$

$\textsc{Find}(x)$:
  return $\text{leader}(x)$

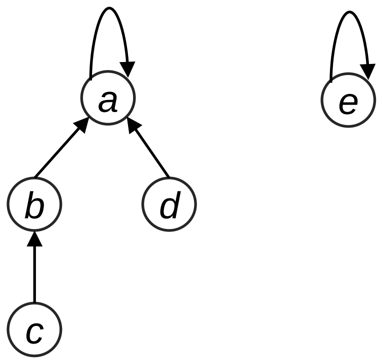
*Sameina*(*x*, *y*) getur tekið *O*(*n*) tíma

# Öfug tré (*reversed trees*, *up trees*)

- Hver hnútur hefur aðeins foreldrabendi, ekki bendi á börn!

- Gerum ráð fyrir að við höfum beinan aðgang í hnútana
  - Til dæmis með vísum í fylki

Rótin bendir á sjálfa sig og er aðalstak (*leader*) mengisins




Tvö tré:

Annað með 4 hnútum, hitt með 1 hnúti

Í upphafi eru öll stökin gerð að eins-staks mengjum, svo eru sum þeirra sameinuð

# Útfærsla aðgerða með öfugum trjám

$\underline{\textsc{MakeSet}(x)\text{:}}$
$\quad parent(x) \leftarrow x$

$\underline{\textsc{Find}(x)\text{:}}$
$\quad \text{while } x \neq parent(x)$
$\qquad x \leftarrow parent(x)$
$\quad \text{return } x$

$\underline{\textsc{Union}(x, y)\text{:}}$
$\quad \overline{x} \leftarrow \textsc{Find}(x)$
$\quad \overline{y} \leftarrow \textsc{Find}(y)$
$\quad parent(\overline{y}) \leftarrow \overline{x}$

<u>Tími</u>: $O(1)$

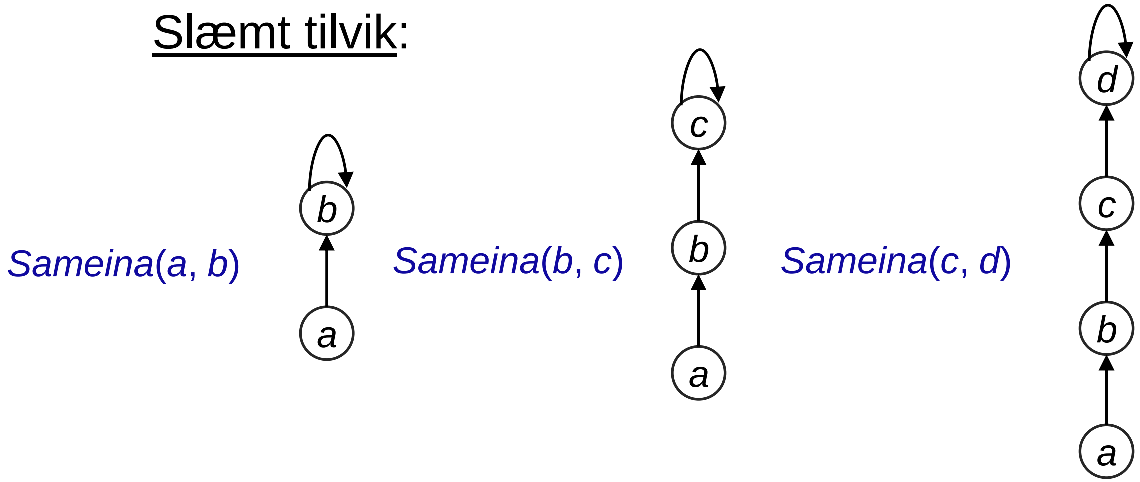
<u>Tími</u>: $O(n)$, því trén geta verið ein löng keðja

<u>Tími</u>: $O(n)$, vegna *Find*-kalla

Byrjum á að búa til mengi úr öllum stökunum

Svo koma *Finna* og *Sameina* aðgerðir

<u>Slæmt tilvik</u>:

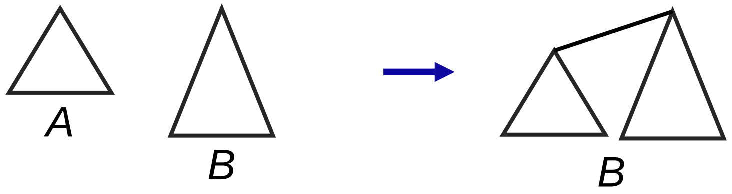


# Endurbætur á *Sameina*

- Ef við vitum dýpi hvers trés þá getum við notað okkur það

Setjum alltaf grynnra tréð sem barn dýpra trésins

Þá eykst dýpið bara þegar bæði trén hafa sama dýpi




$\underline{\text{UNION}(x, y)}$

$\overline{x} \leftarrow \text{FIND}(x)$
$\overline{y} \leftarrow \text{FIND}(y)$
if $depth(\overline{x}) > depth(\overline{y})$
$\quad parent(\overline{y}) \leftarrow \overline{x}$
else
$\quad parent(\overline{x}) \leftarrow \overline{y}$
$\quad$ if $depth(\overline{x}) = depth(\overline{y})$
$\quad\quad depth(\overline{y}) \leftarrow depth(\overline{y}) + 1$

Versta tilfellið er þegar verið er að sameina jafndjúp tré

Minnsti fjöldi staka í þeim trjám er $2^d$, þar sem $d$ er dýpi trjánna

$1, 1 \rightarrow 2 \qquad 2, 2 \rightarrow 4 \qquad 4, 4 \rightarrow 8 \qquad \cdots$

Mesta dýpi $n$-staka trés er þá $\log_2(n)$

Versta tilfellis tími á *Finna* og *Sameina* er nú $O(\log(n))$

# Æfingar

- Höfum stökin *a*, *b*, *c*, *d*.  Sýnið tréð eftir aðgerðirnar:
  - *Sameina*(*a*, *b*), *Sameina*(*b*, *c*), *Sameina*(*c*, *d*)

- Finnið verri röð til að sameina öll stökin í eitt mengi
  - þ.e. röð sem gefur dýpra lokatré

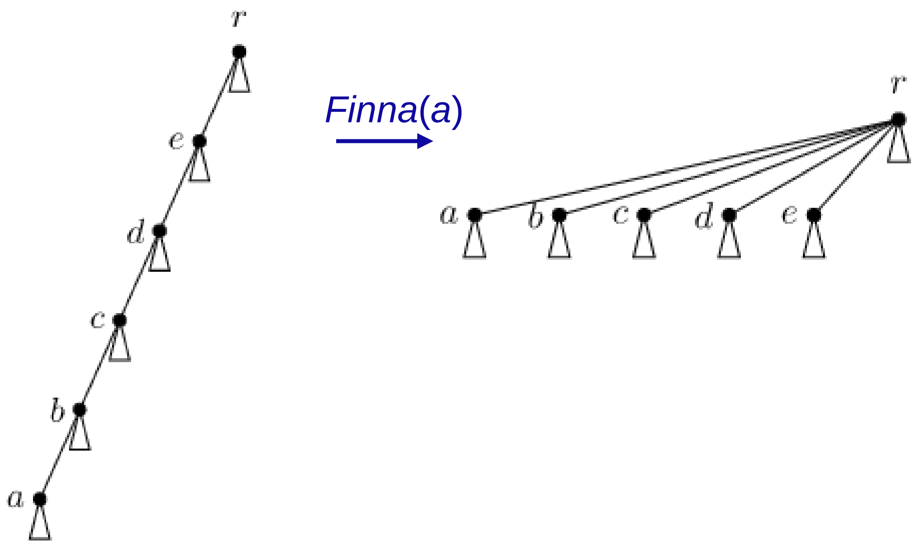
# Vegþjöppun (*path compression*)

- Getum endurbætt *Finna*($x$) aðferðina:

Þessir hnútar eru allir forfeður $x$ í trénu

Þegar við förum upp tréð í leit að rótinni þá geymum við alla hnúta sem við förum framhjá (gerist sjálfkrafa í endurkvæmni)

Þegar rótin er fundin þá látum við alla geymda hnúta benta á rótina



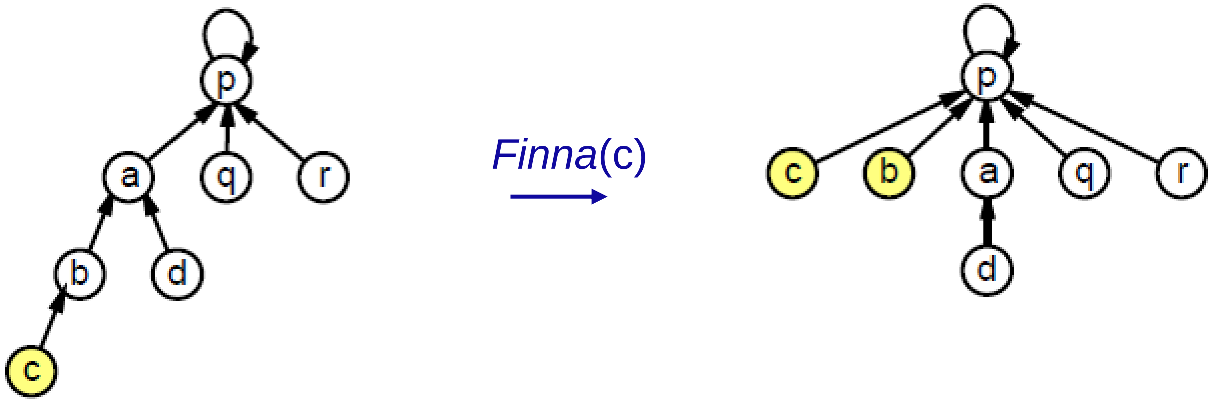

Hættum þegar komin í rót
($x$ er foreldri sjálfs síns)

Nýtt reiknirit:

```
FIND(x)
   if x ≠ parent(x)
         parent(x) ← FIND(parent(x))
   return parent(x)
```

Öll endurkvæmu köllin skila sama gildinu, sem er rótin

# Sýnidæmi um vegþjöppun




Nú verða allar *Finna* aðgerðir á *x* og alla forfeður hans hraðvirkar

Ef ein Finna-aðgerð er dýr, þá verða margar aðrar
Finna-aðgerðir ódýrar í framhaldinu

Getum þá dreift kostnaðinum af dýru
aðgerðinni á margar aðrar aðgerðir!

# Aðrar aðgerðir

Gæti verið mun minna en fjöldi hnúta segir til um

- Athugið að nú verður dýpi trjánna ekki rétt
- Köllum það í staðinn <u>stig</u> (*rank*)

Sameiningin er þá <u>Sameining eftir stigi</u> (*Union by rank*)

<u>$\textsc{MakeSet}(x)$:</u>
    $\mathit{parent}(x) \leftarrow x$
    $\mathit{rank}(x) \leftarrow 0$

<u>$\textsc{Union}(x, y)$</u>
    $\bar{x} \leftarrow \textsc{Find}(x)$
    $\bar{y} \leftarrow \textsc{Find}(y)$
    if $\mathit{rank}(\bar{x}) > \mathit{rank}(\bar{y})$
        $\mathit{parent}(\bar{y}) \leftarrow \bar{x}$
    else
        $\mathit{parent}(\bar{x}) \leftarrow \bar{y}$
        if $\mathit{rank}(\bar{x}) = \mathit{rank}(\bar{y})$
            $\mathit{rank}(\bar{y}) \leftarrow \mathit{rank}(\bar{y}) + 1$

Versta tilfellis tími *Finna* er ennþá $O(\log(n))$

En jafnaðartíminn ætti að vera betri!

Hversu miklu betri er erfitt að sýna

# Jafnaðargreining

- Hægt að sýna að jafnaðartími *Finna-aðgerðarinnar* er $O(\log^*(n))$

log*(*n*) er ítraði logrinn af *n*:

Hversu oft þarf að taka logra áður en útkoman er ≤ 1

$$\log^*(n) = \begin{cases} 1 & \text{ef } n \le 2 \\ 1 + \log^*(\log(n)) & \text{annars} \end{cases}$$

Dæmi:

log*(65536) = 4, vegna þess að

$$\text{4 sinnum} \begin{cases} \log(65536) = 16 & \text{vegna } 2^{16} = 65536 \\ \log(16) = 4 & \text{vegna } 2^4 = 16 \\ \log(4) = 2 & \text{vegna } 2^2 = 4 \\ \log(2) = 1 & \text{vegna } 2^1 = 2 \end{cases}$$

Athugið:

Þetta fall vex **gríðarlega** hægt!

$\log^*(n) \le 5$, fyrir öll $n \le 2^{65536}$

~ $2*10^{19728}$

Fjöldi atóma í hinum sýnilega alheimi er talin vera $10^{80}$

Getum því sagt að $\log^*(n)$ sé ≤ 5 fyrir öll raunhæf gildi á *n*

# Aðeins meira um jafnaðargreininguna

- Nokkrir eiginleikar sem tengjast stigi (*rank*) hnúta:

  Í hvert sinn sem aðalstak *x* breytist þá hefur nýja aðalstakið hærra stig en gamla aðalstakið

  $stærð(\bar{x}) \geq 2^{stig(\bar{x})}$  Fjöldi hnúta í tré með aðalstak $\bar{x}$ er meiri en $2^{stig(\bar{x})}$

  Hæsta mögulega stig í tré með *n* stök er $\lfloor \log_2 n \rfloor$

- Andhverfa Ackermanns fallsins

  Vex hraðar en öll frumstæð rakin föll (*primitive recursive*)

  Ackermann fallið er **ótrúlega hratt vaxandi** fall

  Vex **miklu** hægar en log*

  Andhverfa þess, fallið *α*(*n*) er þá **ótrúlega hægt vaxandi**

  Hægt að sýna að jafnaðartími *Finna-aðgerða* er *O*(*α*(*n*))

  Yfir *m Finna*-aðgerðir á *n* stök

Jafnaðartími *Finna* er því **ekki fasti**, en eins nálægt því og hægt er að vera!

# Aðrar vegþjappanir

Geta orðið $\log_2(n)$ hnútar

- Til þess að losna við að geyma allar hnútana á leiðinni upp í rótina:

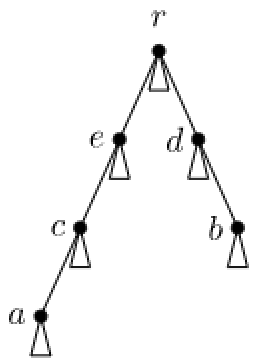
- Vegfleygun (*path split*)

  Á leiðinni upp tréð tengjum við hvern hnút við afa hans

  Þá skiptist slóðin upp í rótina í tvennt

  Hvor um sig helmingurinn að upphaflegu lengdinni

```
Find-ps(x)
   while x != parent(x)
      next = parent(x)
      parent(x) = parent(next)
      x = next
```

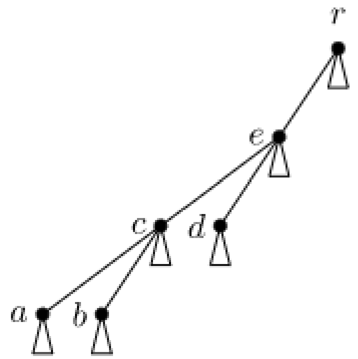



- Veghelmingun (*path halving*)

  Á leiðinni upp tréð tengjum við annan hvern hnút við afa hans

  Fáum þá helmingun á dýpi allra forfeðra x

```
Find-ph(x)
   while x != parent(x)
      parent(x) = parent(parent(x))
      x = parent(x)
```




Hermun á Gnarley trees

# Notkunardæmi

- Að búa til völundarhús (*maze*)

  Byrjum með *nxn* grind af hólfum

  Festum upphafs- og endahólf

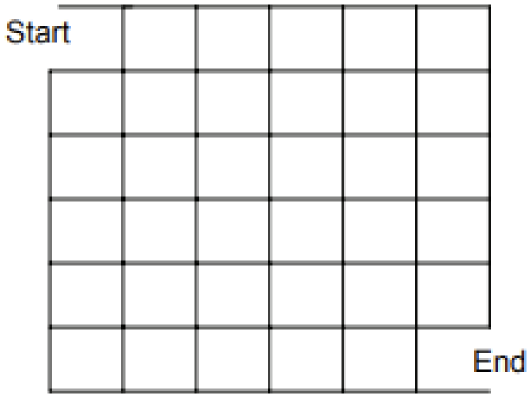
  Skilgreinum hvert hólf sem aðskilið mengi

  Veljum svo af handahófi innri vegg

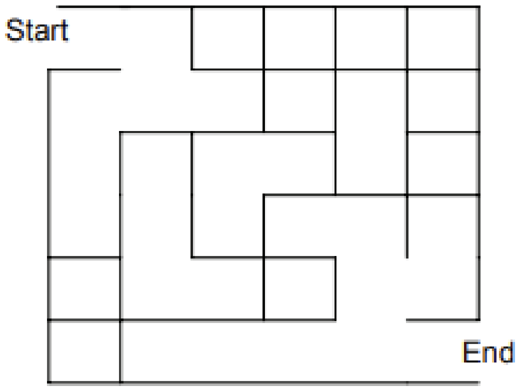
  Ef hólfin sitthvoru megin við vegginn eru í ólíkum mengjum
  þá eyðum við þessum vegg og sameinum mengin

  Veljum svo aftur vegg af handahófi,
  á meðan það eru fleiri en eitt mengi

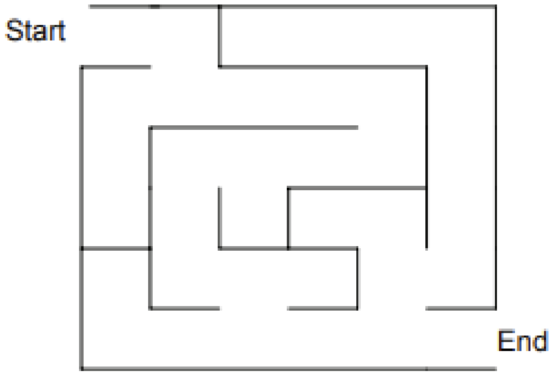
Kemur í veg fyrir að hringir myndist




Upphafsstaða




Millistaða




Lokastaða

# Fyrirlestraæfingar

1. Hver væri tímaflækjan á aðgerðunum *Finna* og *Sameina* ef aðskildu mengin væru útfærð með fylki (þ.a. $A[x]$ inniheldur númerið á aðalstakinu á menginu sem $x$ er í)?

2. Við höfum 1 milljón staka aðskilin mengi sem útfærð eru með vegþjöppun (*path compression*) og sameiningu eftir stigi (*union-by-rank*). Hver er mesti fjöldi hnúta sem breytir um foreldri í einni *Finna* aðgerð?

3. Hversu mörgum veggjum munum við hafa eytt þegar völundarhúsið er tilbúið (þ.e. þegar aðeins eitt mengi er eftir)?