TÖL403G GREINING REIKNIRITA

18. Jafnaðargreining 2

Hjálmtýr Hafsteinsson
Vor 2022

Í þessum fyrirlestri

- Samsettur teljari með hækkun og lækkun
  - Virkni
  - Jafnaðargreining

DC 9.3

- Blórabögglustré (*scapegoat tree*)
  - Leitun
  - Eyðing
  - Innsetning

DC 10.1 − 10.4

Samsettir teljarar

- Sáum síðast samsetta tvíundarteljara

Teljarinn er táknaður með tveimur bitastrengjum (P, N)

Gildi teljarans er $P - N$

Eina skilyrðið sem P og N verða að uppfylla er að $\underline{P \wedge N = 0}$

P og *N* OG-aðar bita fyrir bita eru 00...00

eða: Það má aldrei nema annar af bitunum $P[i]$ og $N[i]$ vera 1

<u>Dæmi</u>:

(1001, 0100) táknar 5, því 9−4 = 5

(0100, 1001) táknar −5, því 4−9 = −5

Getum nú líka táknað neikvæðar tölur

Hækkun og lækkun

- Höfum nú föllin *Increment* og *Decrement*:

$\textsc{Increment}(P,N)$:
$i \leftarrow 0$
while $P[i] = 1$
  $P[i] \leftarrow 0$
  $i \leftarrow i + 1$
if $N[i] = 1$
  $N[i] \leftarrow 0$
else
  $P[i] \leftarrow 1$

$\textsc{Decrement}(P,N)$:
$i \leftarrow 0$
while $N[i] = 1$
  $N[i] \leftarrow 0$
  $i \leftarrow i + 1$
if $P[i] = 1$
  $P[i] \leftarrow 0$
else
  $N[i] \leftarrow 1$

$\textsc{Increment}(P,N)$:
$i \leftarrow 0$
while $P[i] = 1$
  $P[i] \leftarrow 0$
  $i \leftarrow i + 1$
if $N[i] = 1$
  $N[i] \leftarrow 0$
else
  $P[i] \leftarrow 1$

Alveg eins og í upphaflega *Increment*

Verðum að athuga hvort *N*-bitinn er 1 áður en við setjum *P*-bitann

Í stað þess að hækka *P* um 1, þá lækkum við það um 3 og lækkum *N* um 4

<u>Dæmi</u>:

$P = 1001$
$N = 0100$
$P - N = 5$

Inc →

$P = 1010$
$N = 0100$
$P - N = 6$

Hækkum bara *P*

Inc →

$P = 1011$
$N = 0100$
$P - N = 7$

Hækkum bara *P*

Inc →

$P = 1000$
$N = 0000$
$P - N = 8$

Breyting:

$(P - 3) - (N - 4)$
$= P - 3 - N + 4$
$= P - N + 1$

Lækkun

- Lækkun virkar svipað, nema víxlað á *P* og *N*:

```
DECREMENT(P, N):
  i ← 0
  while N[i] = 1
      N[i] ← 0
      i ← i + 1

  if P[i] = 1
      P[i] ← 0
  else
      N[i] ← 1
```

<u>Dæmi</u>:

$$\begin{array}{l} P = 0110 \\ N = 1001 \\ P - N = -3 \end{array} \xrightarrow{Dec} \begin{array}{l} P = 0100 \\ N = 1000 \\ P - N = -4 \end{array}$$

Getum ekki hækkað *N* um 1, svo við lækkum það um 1 og lækkum *P* um 2

<u>Almennt</u>:

Ef <u>hækkun</u> veldur því að bitar $P[k]$ og $N[k]$ verða báðir 1 þá:

Lækkum við *P* um 2^k-1 og lækkum *N* um 2^k

$$\text{Svo } (P - (2^k-1)) - (N - 2^k) = P - \cancel{2^k} + 1 - N + \cancel{2^k}$$
$$= P - N + 1$$

Samskonar fyrir <u>lækkun</u>,
nema víxlað á *P* og *N*

Æfingar

- Hvert er gildi samsetta teljarans (0010, 1100)?

- Lækkið teljarann að ofan tvisvar sinnum

Jafnaðargreining á samsettum teljurum

- Við byrjum í (0, 0) og framkvæmum n *Increment*/*Decrement* aðgerðir

Versta tilfellistími einnar stakrar aðgerðar er $O(\log(n))$ eins og áður

Við vitum ekki hvaða aðgerðir verða framkvæmdar og í hvaða röð

Til dæmis: Ef runan er *Inc*, *Dec*, *Inc*, *Dec*, ... *Inc*, *Dec* — Þá verður gildi teljarans aldrei mjög hátt

Ef runan er *Inc*, *Inc*, ..., *Inc*, *Dec*, *Dec*, ..., *Dec* — Þá fer gildið upp í $n/2$

Gætum líka haft eingöngu *Inc*-aðgerðir eða eingöngu *Dec*-aðgerðir

Það er því erfitt að nota summuaðferðina til að greina kostnaðinn

Þá þarf að leggja saman kostnaðinn við sérhverja aðgerð

Kostnaður einnar aðgerðar er háður fyrri aðgerðum

Skattlagningargreining

- Það er hægt að sýna með þrepun:

Eftir að $P[i]$ eða $N[i]$ hefur verið settur sem 1 þá þarf að minnsta kosti 2^i aðgerðir af *Inc*/*Dec* áður en hann er settur aftur sem 0

Gætu verið fleiri en 2^i aðgerðir

Dæmi:

P = 1011
N = 0000

$\xrightarrow{Inc}$

P = 1100
N = 0000

Þyrftum nú 4 (=2^2) *Inc* aðgerðir til að setja $P[2]$ aftur sem 0

4 (=2^2) *Dec* aðgerðir myndu líka setja $P[2]$ aftur sem 0

En blanda af *Inc*/*Dec* aðgerðum myndi taka lengri tíma að setja $P[2]$ aftur sem 0

Athugið að þegar $P[2]$ er settur sem 1, þá er það vegna þess að $P[0..1]$ = 11, svo þá hlýtur $N[0..1]$ að vera 00

Gætum aldrei hafa verið með: P = 1011, N = 0011

Skattlagning, frh.

- Látum hvern bita $P[i]$ og $N[i]$ borga $1/2^i$ í skatt fyrir hverja aðgerð

Þá eigum við alltaf nóg til að borga fyrir næstu aðgerð

Reyndar meira en nóg!

Til dæmis:

$$\cdots \quad \frac{1}{8} \quad \frac{1}{4} \quad \frac{1}{2} \quad \frac{1}{1}$$

P: ...00110

N: ...00001

$$\cdots \quad \frac{1}{8} \quad \frac{1}{4} \quad \frac{1}{2} \quad \frac{1}{1}$$

Athugið að þessi skattur er bara bókhaldsaðgerð − ekki hluti af reikniritinu!

Jafnaðarkostnaður hverrar aðgerðar er þá: $\displaystyle\sum_{i \geq 0} 2 \cdot \frac{1}{2^i} = \mathbf{4}$

Getum lækkað þetta með nákvæmari greiningu

Stöðuorkugreining

- Fáum þéttari jafnaðargreiningu með stöðuorkuaðferð

Setjum stöðuorkuna Φ_i sem fjölda 1-bita í P og N eftir i aðgerðir

Höfum eins og áður: $a_i = c_i + \Phi_i - \Phi_{i-1}$

þar sem c_i er raunkostnaður og
$\Phi_i - \Phi_{i-1}$ er mismunur í stöðuorku

$c_i =$ (fjöldi bita breytt úr 0 í 1) + (fjöldi bita breytt úr 1 í 0)

$\Phi_i - \Phi_{i-1} =$ (fjöldi bita breytt úr 0 í 1) − (fjöldi bita breytt úr 1 í 0)

svo

$a_i = c_i + (\Phi_i - \Phi_{i-1}) =$ 2*(fjöldi bita breytt úr 0 í 1)

Hver aðgerð breytir í <u>mesta lagi</u> einum bita úr 0 í 1 ➡ $a_i \le 2$

Jafnaðartími hverrar aðgerðar er því í versta tilfelli 2

Sumar breyta engum, t.d. lækkun á (01, 00) yfir í (0, 0)

Blórabögglulstré (*scapegoat tree*)

- Tvíleitartré sem viðheldur jafnvægi á latan (*lazy*) hátt
- Hagkvæmast er að tvíleitartré séu í jafnvægi
  - Þá verður kostnaður við flestar aðgerðir $O(\log(n))$
- Hægt að viðhalda jafnvægi á ýmsa vegu:
  - AVL tré − munur á hæð barna er ≤ 1
  - Rauð-svart tré − viðheldur jafnvægi með litareglum
  - B-tré − stórir hnútar, mismunandi mikið fylltir
  - Splay tré − notar snúninga til að færa algeng stök upp tré
  - Treap − notar slembiforgang til að fá vænt jafnvægi

Allar þessar gagnagrindur nota ákafa (*eager*) umröðun til að viðhalda jafnvægi − þ.e. eitthvað gert í hverri aðgerð til að passa uppá jafnvægið

Sjáum næst!

Sjáum nú aðra gerð: Aðeins að laga tréð þegar við erum að lenda í vandræðum

Kallast löt (*lazy*) hegðun

Stundum kallað íslenska leiðin!

Blórabögglulstré

- Fyrst sett fram af Arne Andersson árið 1989
- og síðan líka af Galperin og Rivest árið 1993

Kallaði þau "*General balanced trees*"

Bjuggu til nafnið "*Scapegoat trees*"

Skoðum fyrst aðeins Leitun (*search*) og Eyðingu (*delete*)

Byrjum með tvíleitartré í fullkomnu jafnvægi: n hnútar, hæð $\lceil \log_2 n \rceil$

Hugmynd: Eyðum ekki hnútum úr trénu við eyðingu, heldur merkjum við þá sem eydda

Leti!

Leitin þarf þá að passa að skoða merkingar hnútanna, því ef við leitum að lykli *k* og hann er í eyddum hnúti þá á leitin að skila *Fannst ekki*

Tímaflækja

- En nú er leitartíminn ekki lengur fall af fjölda lykla í trénu á hverjum tíma!

Lausn: Þegar helmingur hnútanna í trénu hefur verið merktur sem eyddur
þá endurbyggjum við tréð í fullkomnu jafnvægi með ómerktu hnútunum

Við byrjum með tré
í fullkomnu jafnvægi

Höfum nú: Ef tré hefur *n* ómerkta hnúta þá er heildarfjöldi hnúta í
trénu ≤ 2*n* og leit þarf að skoða $\leq \log_2 n + 1$ lykil

Endurbyggingin kostar *O*(*n*) tíma: Breytum trénu í tengdan lista með snúningum
og breytum svo í tré í fullkomnu jafnvægi

Athugið:

Endurbyggingin er ekki framkvæmd fyrr en
eftir ~*n* eyðingar, svo jafnaðarkostnaður við
endurbyggingu er *O*(1) fyrir hverja eyðingu

Rukkum hverja eyðingu um $1
til að borga fyrir endurbygginguna

Leitun: *O*(log(*n*)) versta tilfellistími

Eyðing: *O*(log(*n*)) jafnaðartími

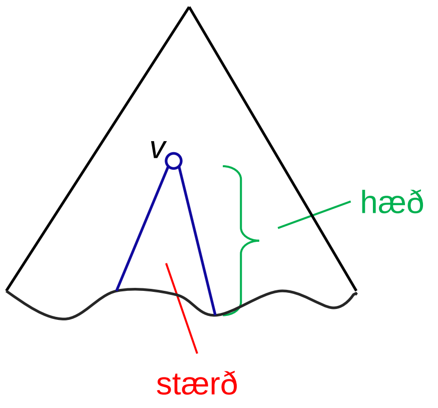
Innsetning

- Setjum ný gildi inn í tréð eins og venjulega, en þegar ójafnvægið <u>er orðið of mikið</u> þá endurbyggjum við hluta trésins

Byggir á stikanum $\alpha > 1$

Hver hnútur v hefur:

hæð(v):	hæð v í trénu, þ.e. fjarlægð frá dýpsta laufi
stærð(v):	fjöldi hnúta í hluttré v (v meðtalið)



<u>Regla</u>:

Eftir innsetningu, fara upp tréð og uppfæra hæð og stærð hnúta á leiðinni. Ef við rekumst á hnút v með

$$hæð(v) > \alpha \cdot \log_2(stærð(v))$$

þá endurbyggja hluttré v þ.a. það sé í fullkomnu jafnvægi

Kostar $O(stærð(v))$ tíma

Blórabögull

- Með þessari reglu verður hæð trésins $\leq \alpha \cdot \log_2(n)$
  - α er fasti, svo leitartíminn er $O(\log(n))$ í versta tilfelli

Hnúturinn v sem veldur endurbyggingu kallast <u>blórabögull</u> (*scapegoat*) því honum er kennt um ójafnvægið, þó hann valdi því ekki beint!

<u>Athugið</u>:

Innsetningar gætu tekið $\Theta(n)$ tíma, því við gætum þurft að endurbyggja allt tréð!

Þetta gerist þó alls ekki alltaf − það þarf að byggjast upp ójafnvægi í trénu

Hægt að sýna að jafnaðartíminn fyrir innsetningu er $O(\log(n))$

Sýnidæmi

Höfum tré með
3 hnútum og $\alpha = 1.4$

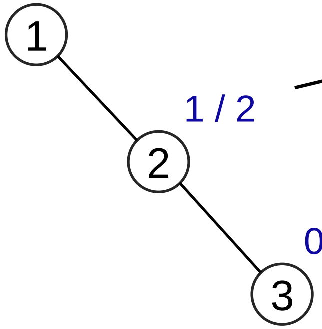
Skoðum hæð og stærð
hvers hnútar

hæð / stærð

2 / 3 — $1.4 \cdot \log_2(3) \approx 2.2$ **≥ 2**

1 / 2 — $1.4 \cdot \log_2(2) = 1.4$ **≥ 1**

0 / 1 — $1.4 \cdot \log_2(1) = 0$ **≥ 0**



Allir hnútar í lagi:

Alls staðar $\alpha \cdot \log_2(\mathit{stærð}(v)) \geq \mathit{hæð}(v)$

Athugum hvort regla uppfyllt

$\alpha \cdot \log_2(\mathit{stærð}(v)) \geq \mathit{hæð}(v)$

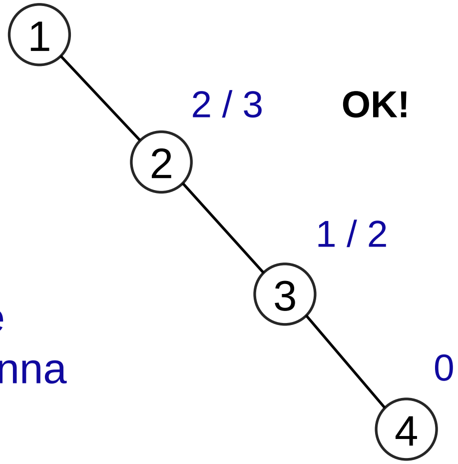
<u>Setjum inn lykilinn 4</u>:

3 / 4 — $1.4 \cdot \log_2(4) = 2.8$ **< 3** Brýtur gegn reglu!

2 / 3 **OK!**

1 / 2 **OK!**

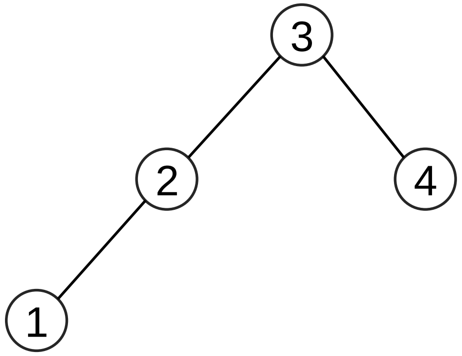
0 / 1 **OK!**



Hnúturinn 1 er nú
blórabögglullinn!

Jafnvel þó ójafnvægið sé
ekki honum einum að kenna

Endurbyggjum
tréð í jafnvægi



Sjá í [Gnarley trees]

Rökstuðningur á jafnaðargreiningu

Skilgreinum: $\textit{ójafnvægi}(v) = |\,\textit{stærð}(\textit{vinstri}(v)) - \textit{stærð}(\textit{hægri}(v))\,|$

mismunur á fjölda hnúta í vinstra og hægra hluttré *v*

Höfum að *ójafnvægi*(*v*) ≤ 1 fyrir alla hnúta *v* í tré í fullkomnu jafnvægi

Getum sýnt að *ójafnvægi*(*v*) = Ω(*stærð*(*v*)) rétt áður en við endurbyggjum *v*

Hver innsetning hækkar ójafnvægið um 1 í mesta lagi

Ef ójafnvægið í *v* er orðið svona mikið (≥ *c*·*stærð*(*v*)) þá hljóta að hafa verið a.m.k. *stærð*(*v*) lyklar settir inn í tréð síðan síðasta endurbygging

Endurbygging á hnúti *v* kosta *O*(*stærð*(*v*)) tíma, svo við getum skipt þeim kostnaði á *stærð*(*v*) innsetningar. Hver þeirra rukkuð um *O*(1) fyrir það

Hver hnútur fer inn í mest *O*(log(*n*)) hluttré

Forfeður hans í trénu

Þetta þýðir að jafnaðartími á Innsetningu er *O*(log(*n*))

Samantekt

- Eiginleikar blórabögglustrjáa:

Leitun:	$O(\log(n))$ tími í versta tilfelli
Innsetning:	$O(\log(n))$ jafnaðartími
Eyðing:	$O(\log(n))$ jafnaðartími

Munið:

Jafnaðartími er versta tilfellistími fyrir runu aðgerða (deilt á aðgerðirnar)

Versta tilfellistími er mesti tími sem ein stök aðgerð getur tekið

Sterkari mörk

- Einfaldanir:
  - Hægt að sýna að það verður í mesta lagi einn blóraböggull í hverri innsetningu
  - Hægt að komast hjá því að geyma hæð og stærð í hverjum hnúti

    Getum reiknað þau gildi við innsetningu með viðbótarkostnaði sem er $O(1)$

Flestar aðrar tegundir af tvíleitartré í jafnvægi þurfa að geyma einhverja jafnvægisupplýsingar í hverjum hnúti

Fyrirlestraæfingar

1. Framkvæmið *Lækka* á samsetta teljaranum (1010, 0101) (sem er (10, 5), þ.e. $10-5=5$)

2. Hvaða röð á aðgerðunum *Hækka* / *Lækka* gefur okkur samsetta teljarann (100, 011)?

3. Við eyðum lykli *k* innan úr miðju blórabögglstrés og setjum gildið *k* síðan aftur inn, áður en tréð hefur verið endurbyggt. Hvar kemur þetta nýja gildi: á sama stað og það var, eða sem lauf í trénu?