

Lenivé haldy

kuko

7.10.2020

Vybrané partie z datových štruktúr

binom. halda

merge	$O(\log n)$
insert	$O(\log n)$
extract-min	$O(\log n)$
decrease-key	$O(\log n)$

	binom. halda	
merge	$O(\log n)$	$O(1)$ am.
insert	$O(\log n)$	$O(1)$ am.
extract-min	$O(\log n)$	$O(\log n)$ am.
decrease-key	$O(\log n)$	$O(\log n)$

	binom. halda	lenivá bin. halda
merge	$O(\log n)$	$O(1)$ am.
insert	$O(\log n)$	$O(1)$ am.
extract-min	$O(\log n)$	$O(\log n)$ am.
decrease-key	$O(\log n)$	$O(\log n)$

	binom. halda	lenivá bin. halda	
merge	$O(\log n)$	$O(1)$ am.	$O(1)$ am.
insert	$O(\log n)$	$O(1)$ am.	$O(1)$ am.
extract-min	$O(\log n)$	$O(\log n)$ am.	$O(\log n)$ am.
decrease-key	$O(\log n)$	$O(\log n)$	$O(1)$ am.

	binom. halda	lenivá bin. halda	Fibonacciho halda
merge	$O(\log n)$	$O(1)$ am.	$O(1)$ am.
insert	$O(\log n)$	$O(1)$ am.	$O(1)$ am.
extract-min	$O(\log n)$	$O(\log n)$ am.	$O(\log n)$ am.
decrease-key	$O(\log n)$	$O(\log n)$	$O(1)$ am.

Lenivá binomiálna halda

INVARIANT: Každý koreň stromu má 1\$.

- merge: 1\$
- insert: 1\$
- extract-min: $2 \log n + 1$ \$

INVARIANT: Každý koreň stromu má 1\$.

- merge: 1\$
- insert: 1\$
- extract-min: $2 \log n + 1$ \$

analýza extract-min:

- zmažeme min a pripojíme deti (1\$)
- každé dieťa sa stane nový koreň \Rightarrow dostane 1\$ (spolu $\leq \log n$)
- upratanie t stromov trvá $O(t)$; rozdelíme stromy na dve skupiny:
 - $\ell = \#$ stromov, ktoré prilinkujeme pod iné
 - $r = \#$ koreňov, ktoré ostanú koreňmi aj po upratovaní
- korene, ktoré prilinkujeme si to sami zaplatia
- $r \leq \log n$ – toto zaplatíme z peňazí za operáciu

analýza extract-min:

- zmažeme min a pripojíme deti (1\$)
- každé dieťa sa stane nový koreň \Rightarrow dostane 1\$ (spolu $\leq \log n$)
- upratanie t stromov trvá $O(t)$; rozdelíme stromy na dve skupiny:
 - $\ell = \#$ stromov, ktoré prilinkujeme pod iné
 - $r = \#$ koreňov, ktoré ostanú koreňmi aj po upratovaní
- korene, ktoré prilinkujeme si to sami zaplatia
- $r \leq \log n$ – toto zaplatíme z peňazí za operáciu

analýza extract-min:

- zmažeme min a pripojíme deti (1\$)
- každé dieťa sa stane nový koreň \Rightarrow dostane 1\$ (spolu $\leq \log n$)
- upratanie t stromov trvá $O(t)$; rozdelíme stromy na dve skupiny:
 - $\ell = \#$ stromov, ktoré prilinkujeme pod iné
 - $r = \#$ koreňov, ktoré ostanú koreňmi aj po upratovaní
- korene, ktoré prilinkujeme si to sami zaplatia
- $r \leq \log n$ – toto zaplatíme z peňazí za operáciu

analýza extract-min:

- zmažeme min a pripojíme deti (1\$)
- každé dieťa sa stane nový koreň \Rightarrow dostane 1\$ (spolu $\leq \log n$)
- upratanie t stromov trvá $O(t)$; rozdelíme stromy na dve skupiny:
 - $\ell = \#$ stromov, ktoré prilinkujeme pod iné
 - $r = \#$ koreňov, ktoré ostanú koreňmi aj po upratovaní
- korene, ktoré prilinkujeme si to sami zaplatia
- $r \leq \log n$ – toto zaplatíme z peňazí za operáciu

analýza extract-min:

- zmažeme min a pripojíme deti (1\$)
- každé dieťa sa stane nový koreň \Rightarrow dostane 1\$ (spolu $\leq \log n$)
- upratanie t stromov trvá $O(t)$; rozdelíme stromy na dve skupiny:
 - $l = \#$ stromov, ktoré prilinkujeme pod iné
 - $r = \#$ koreňov, ktoré ostanú koreňmi aj po upratovaní
- korene, ktoré prilinkujeme si to sami zaplatia
- $r \leq \log n$ – toto zaplatíme z peňazí za operáciu

analýza extract-min:

- zmažeme min a pripojíme deti (1\$)
- každé dieťa sa stane nový koreň \Rightarrow dostane 1\$ (spolu $\leq \log n$)
- upratanie t stromov trvá $O(t)$; rozdelíme stromy na dve skupiny:
 - $l = \#$ stromov, ktoré prilinkujeme pod iné
 - $r = \#$ koreňov, ktoré ostanú koreňmi aj po upratovaní
- korene, ktoré prilinkujeme si to sami zaplatia
- $r \leq \log n$ – toto zaplatíme z peňazí za operáciu

analýza extract-min:

- zmažeme min a pripojíme deti (1\$)
- každé dieťa sa stane nový koreň \Rightarrow dostane 1\$ (spolu $\leq \log n$)
- upratanie t stromov trvá $O(t)$; rozdelíme stromy na dve skupiny:
 - $l = \#$ stromov, ktoré prilinkujeme pod iné
 - $r = \#$ koreňov, ktoré ostanú koreňmi aj po upratovaní
- korene, ktoré prilinkujeme si to sami zaplatia
- $r \leq \log n$ – toto zaplatíme z peňazí za operáciu

analýza extract-min:

- zmažeme min a pripojíme deti (1\$)
- každé dieťa sa stane nový koreň \Rightarrow dostane 1\$ (spolu $\leq \log n$)
- upratanie t stromov trvá $O(t)$; rozdelíme stromy na dve skupiny:
 - $l = \#$ stromov, ktoré prilinkujeme pod iné
 - $r = \#$ koreňov, ktoré ostanú koreňmi aj po upratovaní
- korene, ktoré prilinkujeme si to sami zaplatia
- $r \leq \log n$ – toto zaplatíme z peňazí za operáciu

$$\Phi(H) = \# \text{bin. stromov} = \# \text{koreňov}$$

operácia `extract-min` pre haldu s t koreňmi trvá $O(t + \log n)$
 $= O(\ell + \log n)$ zmena potenciálu $\Delta\Phi \leq \log n - \ell$

$$T_{amort} = T_{skut} + c \cdot \Delta\Phi = O(\ell + \log n) + c \cdot (\log n - \ell) = O(\log n)$$

$$\Phi(H) = \# \text{bin. stromov} = \# \text{koreňov}$$

operácia `extract-min` pre haldu s t koreňmi trvá $O(t + \log n)$
 $= O(\ell + \log n)$ zmena potenciálu $\Delta\Phi \leq \log n - \ell$

$$T_{amort} = T_{skut} + c \cdot \Delta\Phi = O(\ell + \log n) + c \cdot (\log n - \ell) = O(\log n)$$

$$\Phi(H) = \# \text{bin. stromov} = \# \text{koreňov}$$

operácia `extract-min` pre haldu s t koreňmi trvá $O(t + \log n)$
 $= O(\ell + \log n)$ zmena potenciálu $\Delta\Phi \leq \log n - \ell$

$$T_{amort} = T_{skut} + c \cdot \Delta\Phi = O(\ell + \log n) + c \cdot (\log n - \ell) = O(\log n)$$

$$\Phi(H) = \# \text{bin. stromov} = \# \text{koreňov}$$

operácia `extract-min` pre haldu s t koreňmi trvá $O(t + \log n)$
 $= O(\ell + \log n)$ zmena potenciálu $\Delta\Phi \leq \log n - \ell$

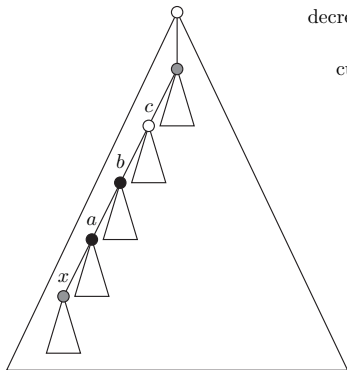
$$T_{amort} = T_{skut} + c \cdot \Delta\Phi = O(\ell + \log n) + c \cdot (\log n - \ell) = O(\log n)$$

$$\Phi(H) = \# \text{bin. stromov} = \# \text{koreňov}$$

operácia `extract-min` pre haldu s t koreňmi trvá $O(t + \log n)$
 $= O(\ell + \log n)$ zmena potenciálu $\Delta\Phi \leq \log n - \ell$

$$T_{amort} = T_{skut} + c \cdot \Delta\Phi = O(\ell + \log n) + c \cdot (\log n - \ell) = O(\log n)$$

Fibonacciho halda



decrease key of x by δ

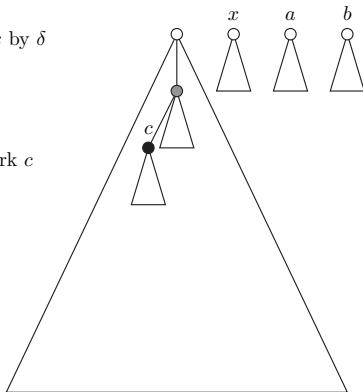
\implies

cut x

\rightarrow cut a

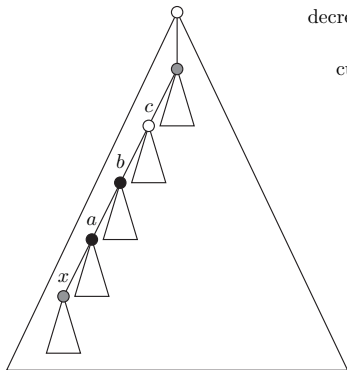
\rightarrow cut b

\rightarrow mark c



INVARIANT: každý označený vrchol, ktorý prišiel o syna, má nasporené 2\$

na decrease-key stačia 4\$:



decrease key of x by δ

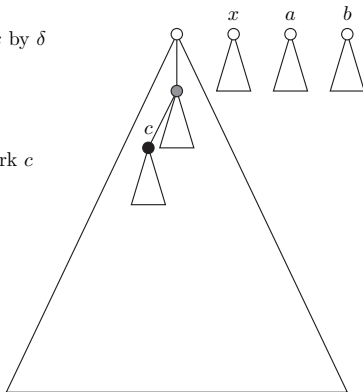
\implies

cut x

\rightarrow cut a

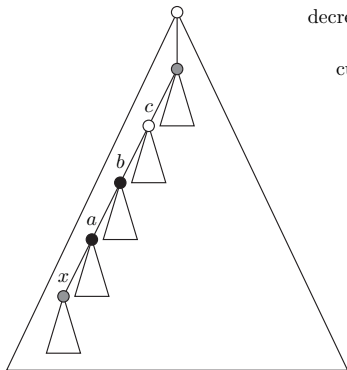
\rightarrow cut b

\rightarrow mark c



INVARIANT: každý označený vrchol, ktorý prišiel o syna, má nasporené 2\$

na decrease-key stačia 4\$:



decrease key of x by δ

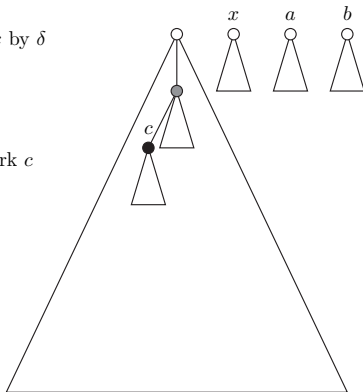
\implies

cut x

→ cut a

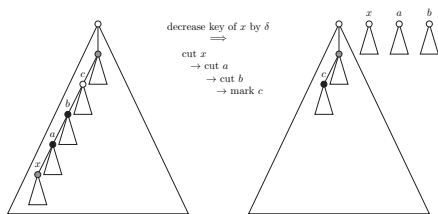
→ cut b

→ mark c



INVARIANT: každý označený vrchol, ktorý prišiel o syna, má nasporené 2\$

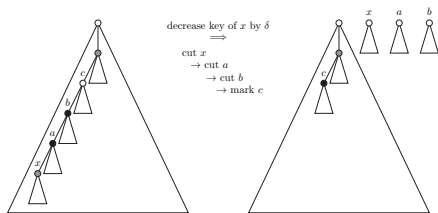
na decrease-key stačia 4\$:



$$\Phi(H) = \#\text{koreňov} + 2 \times \#\text{označených vrcholov}$$

ak odrežeme k vrcholov, trvá to $O(k)$, ale potenciál klesne o $k - 4$

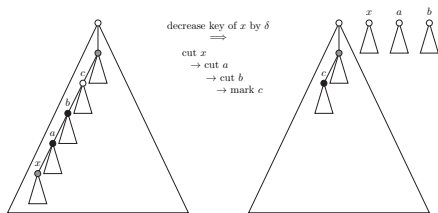
- $+k$ kvôli novým koreňom
- $-2 \times (k - 1)$, lebo všetky okrem prvého koreňa boli označené a teraz nie sú
- $+2$ jeden označený na konci kaskády pribudol



$$\Phi(H) = \#\text{koreňov} + 2 \times \#\text{označených vrcholov}$$

ak odrežeme k vrcholov, trvá to $O(k)$, ale potenciál klesne o $k - 4$

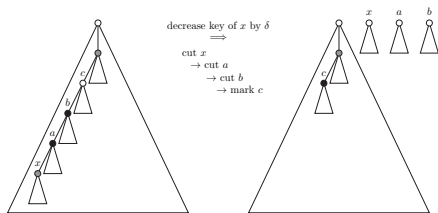
- $+k$ kvôli novým koreňom
- $-2 \times (k - 1)$, lebo všetky okrem prvého koreňa boli označené a teraz nie sú
- $+2$ jeden označený na konci kaskády pribudol



$$\Phi(H) = \# \text{koreňov} + 2 \times \# \text{označených vrcholov}$$

ak odrežeme k vrcholov, trvá to $O(k)$, ale potenciál klesne o $k - 4$

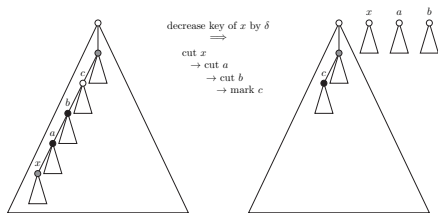
- $+k$ kvôli novým koreňom
- $-2 \times (k - 1)$, lebo všetky okrem prvého koreňa boli označené a teraz nie sú
- $+2$ jeden označený na konci kaskády pribudol



$$\Phi(H) = \#\text{koreňov} + 2 \times \#\text{označených vrcholov}$$

ak odrežeme k vrcholov, trvá to $O(k)$, ale potenciál klesne o $k - 4$

- $+k$ kvôli novým koreňom
- $-2 \times (k - 1)$, lebo všetky okrem prvého koreňa boli označené a teraz nie sú
- $+2$ jeden označený na konci kaskády pribudol



$$\Phi(H) = \# \text{koreňov} + 2 \times \# \text{označených vrcholov}$$

ak odrežeme k vrcholov, trvá to $O(k)$, ale potenciál klesne o $k - 4$

- $+k$ kvôli novým koreňom
- $-2 \times (k - 1)$, lebo všetky okrem prvého koreňa boli označené a teraz nie sú
- $+2$ jeden označený na konci kaskády pribudol

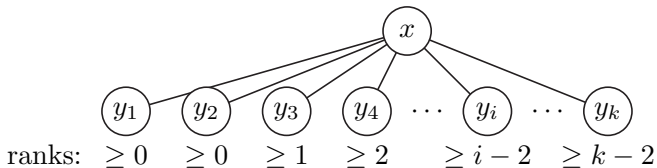
Veta

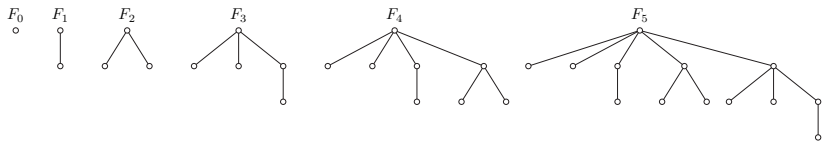
Nech $\text{rank}(v) = \#\text{synov vrcholu}$.

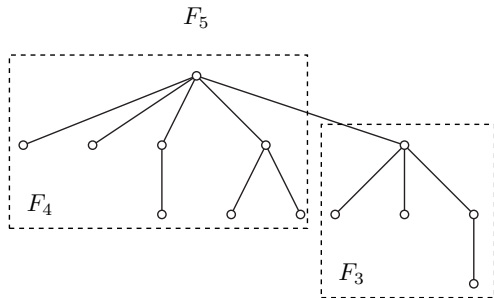
Pri takomto orezávaní bude mať vrchol s rankom k exponenciálne veľa vrcholov v závislosti od k .

Lema

Nech vrchol x má synov y_1, \dots, y_m , ktorých sme označili v poradí, ako sa prilinkovali ku x ; potom $\text{rank}(y_i) \geq i - 2$.







- veľkosť stromu s rankom k je aspoň $F_{k+1} \geq \phi^k$, kde $\phi \approx 1.618$
- \Rightarrow maximálny rank je logaritmický
- \Rightarrow každý koreň má najviac $O(\log n)$ synov a po uprataní ostane najviac $O(\log n)$ stromov

- veľkosť stromu s rankom k je aspoň $F_{k+1} \geq \phi^k$, kde $\phi \approx 1.618$
- \Rightarrow maximálny rank je logaritmický
- \Rightarrow každý koreň má najviac $O(\log n)$ synov a po uprataní ostane najviac $O(\log n)$ stromov

- veľkosť stromu s rankom k je aspoň $F_{k+1} \geq \phi^k$, kde $\phi \approx 1.618$
- \Rightarrow maximálny rank je logaritmický
- \Rightarrow každý koreň má najviac $O(\log n)$ synov a po uprataní ostane najviac $O(\log n)$ stromov

