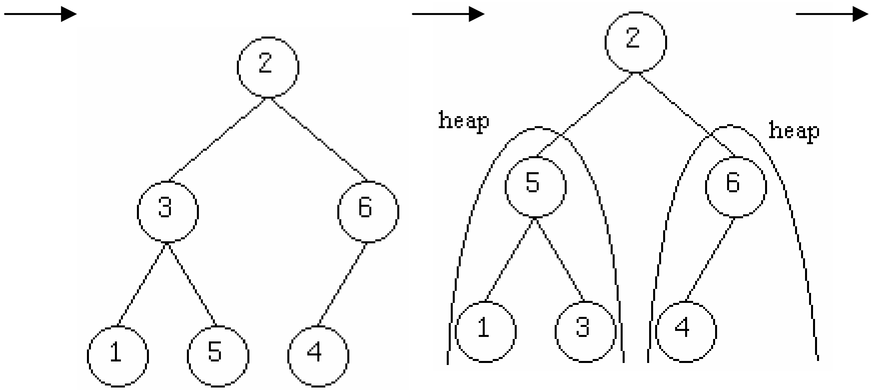
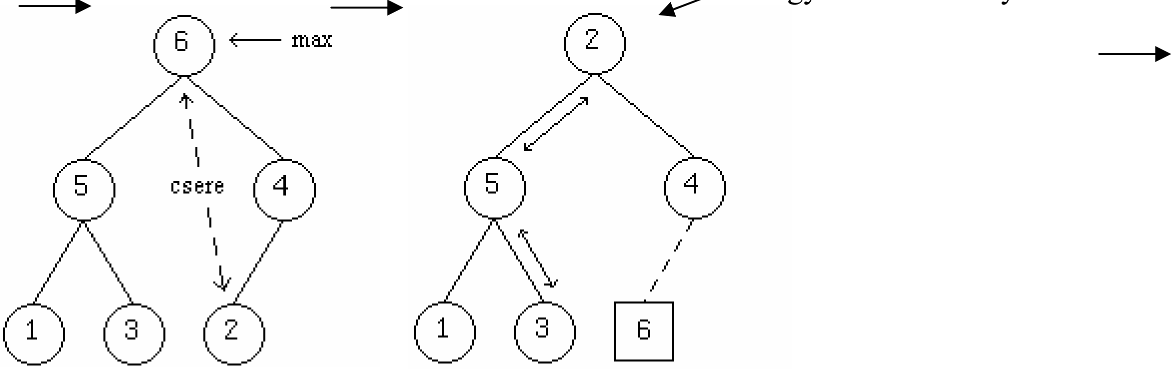
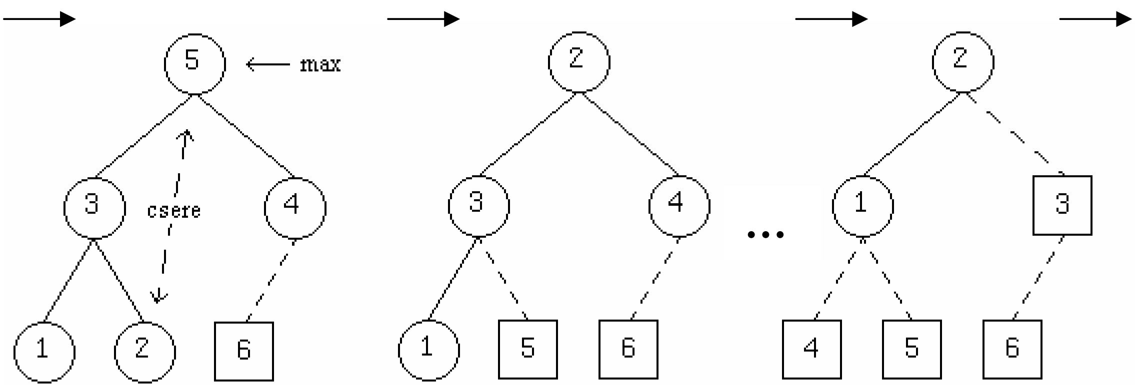
11. Kupac Rendezés
(Heap Sort)

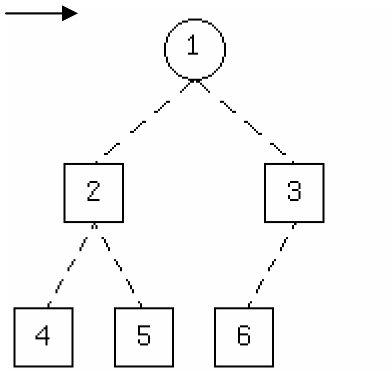
Maximum kiválasztó rendezés.
„A tournament rendező, helyben rendező változata.”
Heap: lásd elsőbbségi sornál
Pl.: 2 3 6 1 5 4

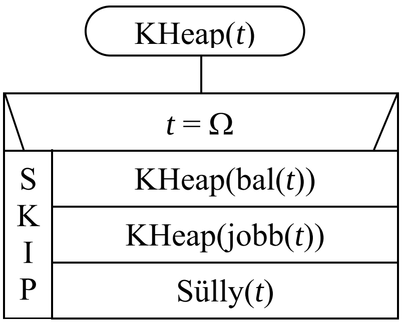


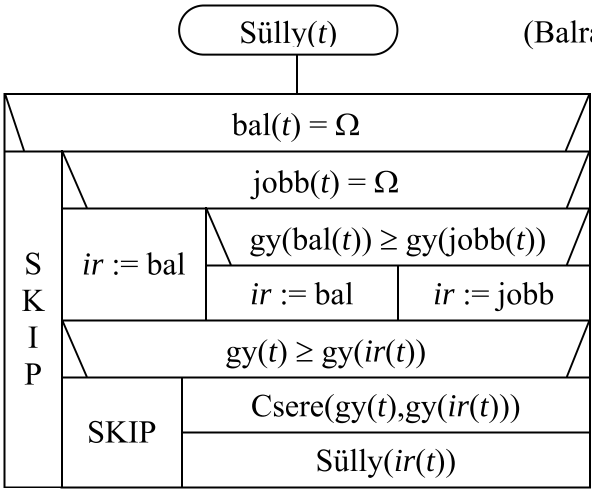
elromlott a heap; ez helyreáll a gyökér elem süllyesztésével





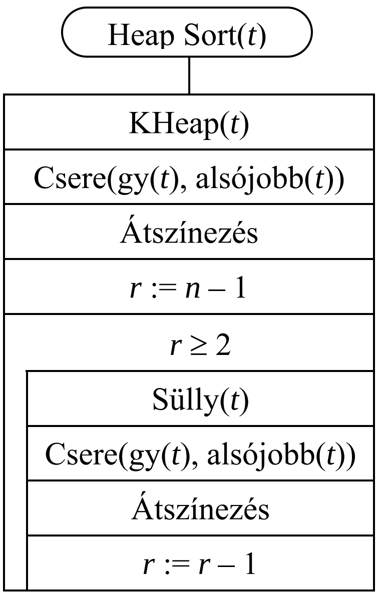






(Balratömörített fa)

$t \neq \Omega$



$t \neq \Omega$, balratömörített fehér fa

Átszínezés: alsójobb(t)-be vezető út átszínezése fehérről pirosra.

σ meghatározható szektorral kicserélhetők (kiválthatók) a színek.

-t fehér

$T(k) = MCs(N) = (2^{k+1} - 1) - k - 1 = N - \lfloor \log_2 N \rfloor - 1$

lineáris, 1-es faktorral

$2^k \le n \le N = 2^{k+1} - 1$ $\quad$ (n=8, 9, 10, …, 15 ; N=15)

$MCs(n) \le MCs(N) = N - \lfloor \log_2 N \rfloor - 1$

Hogyan lehetne a jobb oldalon n függvényéhez jutni?

a) $N < 2n$

$MCs(n) < 2n - \lfloor \log_2 n \rfloor (-1)$ $\qquad$ (újra itt a 2-es faktor!)

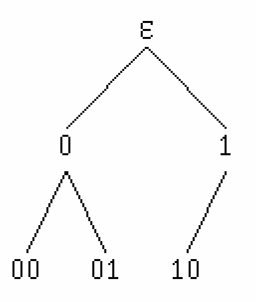
b) „Szemmel látható” (?), hogy elég nagy n-ekre a T(n) értékek „közel vannak” az 1 meredekségű egyeneshez. K=1 tartható?

Az algoritmus más formában: σ alkalmazásával.

Bináris fa meghatározó szelektora: σ=10

szelektorhalmaza:

Sel(t)={ ε, 0, 1, 00, 01, 10}



Rendezés (a szelektorok halmazán): lexikografikus

t balra tömörített (és ennek következtében majdnem teljes $\Leftrightarrow$ $\exists \sigma : Sel(t) = [\varepsilon..\sigma]$

Ekkor σ-t a t meghatározó szelektorának nevezzük.

t $\Leftrightarrow$ σ kölcsönösen egyértelmű kapcsolat.

Legyen σ^- a σ-t megelőző szelektor a rendezettségben. ($\sigma \ne \varepsilon$)

Pl.: 10^-=01

Heap Sort(t_σ)

KHeap(t_σ)
Csere(gy(t_σ),σ(t_σ))
$\sigma := \sigma^-$
$\sigma \ne \varepsilon$
Sülly(t_σ)
Csere(gy(t_σ),σ(t_σ))
$\sigma := \sigma^-$

a KHeap eljárás marad ugyanaz

Sülly(t)

h(t) ≤ 0			
SKIP	j := MaxÉrtékSorsz(gy(0(t)),gy(1(t)),gy(t))		
	$j = 0$	$j = 1$	$j = 2$
	Csere(gy(t), gy(0(t)))	Csere(gy(t), gy(1(t)))	SKIP
	Sülly(0(t))	Sülly(1(t))	

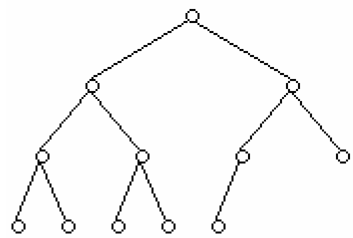
A műveletigény

I. Pl.: n=12, k=3

k=3 $\Leftrightarrow$ n=8, 9, 10, …, 15

k=h(t) $\Leftrightarrow$ $2^k \le n < 2^{k+1}$

$k \le \log_2 \le k+1$



h	# Cs
0	1
1	2
2	4
3	≤8

Az i-edik magasságban lévő elemek száma: $0 \le i < k \Rightarrow 2^i$ elem

$i = k \Rightarrow \le 2^i$ elem

MCs(n)=?

KHeap

$$\text{MCs(n)} \le 1k + 2^1(k-1) + 2^2(k-2) + ... + 2^{k-1}1 = \sum_{i=0}^{k} 2^i (k-i) = \sum_{j=1}^{k} 2^{k-j} j = 2^k \sum_{j=1}^{k} \frac{j}{2^j} < 2^k 2 \le 2n$$

lineáris művelet

1 2 3 … k

$\frac{1}{2}$

$\frac{1}{4}$ $\frac{1}{4}$

$\frac{1}{8}$ $\frac{1}{8}$ $\frac{1}{8}$

… … …

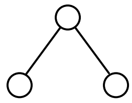
$\frac{1}{2^k}$ $\frac{1}{2^k}$ $\frac{1}{2^k}$ … $\frac{1}{2^k}$

----

1 $\frac{1}{2}$ $\frac{1}{4}$ … $\frac{1}{2^{k-1}}$ < 2

$MÖ(n) < 4n$ (# összehasonítások: # cserék = 2:1)

2 kérdés = 1 csere



Sülly
A gyökér legrosszabb esetben h(t) élet süllyed; ez ennyi csere.
Ha minden elemet k magasságig süllyesztenénk, akkor h(t)= $k = \lfloor \log_2 n \rfloor$.

$MCs(n) < (n-1)\lfloor \log_2 n \rfloor = O(n \log_2 n)$

$MÖ(n) \approx 2MCs(n) = O(n \log_2 n)$

A teljes heapsort-ra: $MCs_{HS}(n) < 2n + (n-1)\lfloor \log_2 n \rfloor = O(n \log_2 n)$

$MÖ_{HS}(n) = O(n \log_2 n)$

II. Rekurzív összefüggés

KHeap
$MCs(t) \le MCs(bal(t)) + MCs(jobb(t)) + h(t)$

h(t)=0 esetén ez 0

Ennek általános megoldása helyett tekintsük a teljes bináris fák N pontszámait:

$N = 2^{k+1} - 1$

Jelölés: T=MCs; T(fa magassága)

T(0)=0
T(1)=1
T(i)=2T(i-1)+i (i>0)
gyökér bevitele

k	N	T	
0	1	0	
1	3	1	4-(1+2)
2	7	4	8-(2+2)
3	15	11	16-(3+2)
4	31	26	32-(4+2)
5	63	57	64(5+2)
k	N	$2^{k+1} - (k+2)$	(Bizonyítandó teljes indukcióval)

Hatékonysági elemzés:
KHeap: $MCs(n) < 2n; MÖ(n) < 4n$

Megjegyzés:

pontos számítás $n = 2^{k+1} - 1$-re
durva becslés tetszőleges n-re $2^k \le n \le 2^{k+1} - 1$; 2-es faktor

Süllyesztés: $MCs(n) \le h(t) \le \lfloor \log_2 n \rfloor$

Ha mind az (n-1) elem ilyen hosszú úton süllyedne, akkor:

$MCs(n) < (n-1)\log_2 n$ lenne.

<u>A teljes heapsort-ra:</u> $MCs(n) < 2n + (n-1)(\log_2 n + 1) = O(n \log_2 n)$

$M\ddot{O}(n) = 2MCs(n) = O(n \log_2 n)$

általános eset:

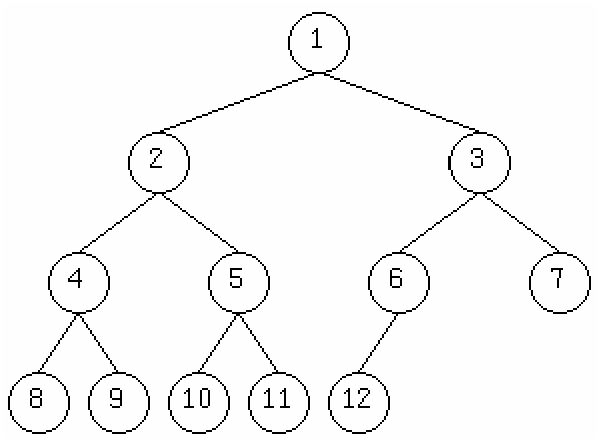
(egy teljes bináris fa elemeinek több, mint a fele levél!)

$\Rightarrow$ átlagos eset fákra $\approx$ legrosszabb eset

$ACs(n) = O(n \log n)$

$A\ddot{O}(n) = O(n \log n)$

A heapsort iteratív változata aritmetikai ábrázolás esetén:



$\rightarrow$

1	2	3	4	5	6	7	8	9	10	11	12

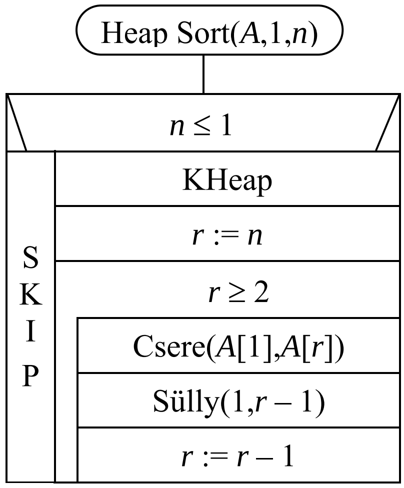
A[u]-tól, mint gyökértől kezdve süllyeszt A[u]-val bezárólag

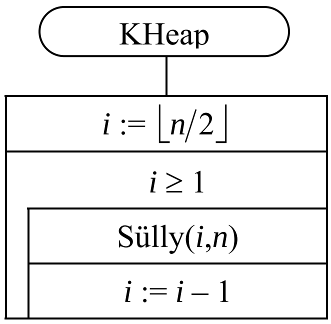
Sülly(A,u,v)

„kell-e süllyeszteni” — l := igaz

a bal gyerek a fához tartozik — $2u \le v \wedge l$

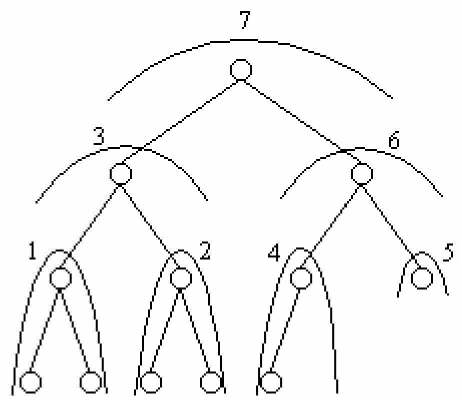
- $2u + 1 > v \vee A[2u] > A[2u + 1]$ — nincs jobb gyerek, vagy a bal gyerek nagyobb
  - igen: $ir := 2u$
  - nem: $ir := 2u + 1$
- $A[u] \ge A[ir]$
  - igen: l := hamis
  - nem: Csere($A[u]$,$A[ir]$); $u := ir$





Az alsó jobb levél szülője

A rekurzív KHeap sorrendje:



Iteratív működési sorrend: