PODATKOVNE STRUKTUKTURE IN ALGORITMI

Vaje

4.10.24

Vaje 1

$O(f(n)) = \{g(n) : (\exists c>0)(\exists n_0 \in \mathbb{N})(\forall n)(n \geq n_0 \Rightarrow g(n) \leq c\cdot f(n)\}$

$o(f(n)) = \{g(n) : (\forall c>0)(\exists n_0 \in \mathbb{N})(\forall n)(n \geq n_0 \Rightarrow g(n) < c\cdot f(n)\}$

$\Omega(f(n)) = \{g(n) : (\exists c>0)(\exists n_0 \in \mathbb{N})(\forall n)(n \geq n_0 \Rightarrow g(n) \geq c\cdot f(n)\}$

$g = \Theta(f(n)) \Leftrightarrow g = O(f(n)) \wedge g = \Omega(f(n))$

Velja: $g(n) = o(f(n)) \Leftrightarrow \lim_{n\to\infty} \frac{g(n)}{f(n)} = 0$

Naloga 1

Algoritem ima časovno zahtevnost $T(n) = \frac{1}{2}n^2 - 3n$.
Pokaži, da je $T(n) = \Theta(n^2)$

a) $\underbrace{T(n)}_{g(x)} = O(\underbrace{n^2}_{f(x)})$

Iščemo $c>0$ in $n_0 \in \mathbb{N}$, da velja
$\frac{1}{2}n^2 - 3n \leq c\cdot n^2$ za $n \geq n_0$.

$\frac{1}{2} - \frac{3}{n} \leq c$

Vzamemo $c = \frac{1}{2}$. Torej $\frac{1}{2} - \frac{3}{n} < \frac{1}{2}$ $\quad \frac{3}{n} > 0$

Vzamemo $n_0 = 1$.

b) $T(n) = \Omega(n^2)$

Iščemo $c > 0$ in $n_0 \in \mathbb{N}$, da za vse $n \geq n_0$ velja

$$\frac{1}{2}n^2 - 3n \geq cn^2$$
$$\frac{1}{2} - \frac{3}{n} \geq c$$

Vzamemo $c = \frac{1}{4}$.

$$\frac{1}{2} - \frac{3}{n} \geq \frac{1}{4} \quad / 4n$$

$2n - 12 \geq 1 \Rightarrow n \geq 12 \Rightarrow$ Vzamemo $n_0 = 12$

Naloga 2

Pokaži, da velja:

a) $2^{n+1} = O(2^n)$

b) $2^{2n} \neq O(2^n)$

a) Iščemo $c > 0$ in $n_0 \in \mathbb{N}$, da za vse $n \geq n_0$ velja:

$$2^{n+1} \leq c2^n$$
$$2 \cdot 2^n \leq c \cdot 2^n$$
$$2 \leq c$$

Vzamemo $c = 2$ in $n_0 = 1$, ker je neodvisno od n.

Opomba: če je $f = O(g)$ potem je $g \neq O(f)$

b) 1. način: po def.

$g \neq O(f(n)) \Leftrightarrow (\forall c > 0)(\forall n_0 \in \mathbb{N})(\exists n)((n \geq n_0) \wedge g(n) > cf(n))$

Naj bosta $c > 0$ in $n_0 \in \mathbb{N}$ poljubna. Iščemo n, da velja $n \geq n_0$ in $2^{2n} > c \cdot 2^n$. Sledi

$$2^{2n} > c2^n \quad / : 2^n$$
$$2^n > c \quad / \log_2$$
$$n > \log_2 c$$

Vzamemo $n = \log_2 c + n_0$
$(n(c, n_0))$

2. način: z limito

$$\lim_{n\to\infty} \frac{2^n}{2^{2n}}$$
$$= \lim_{n\to\infty} \frac{2^n}{2^n \cdot 2^n}$$
$$= \lim_{n\to\infty} \frac{1}{2^n} = \underline{\underline{0}}$$

Naloga 3

Algoritem z izbiro

Vhodni podatki: polje A celih števil z dolžino n.

Izhodni podatki: naraščajoče urejeno polje A

```
for i=1 to n-1 do
  minIndex=i
  for j=i+1 to n
     if A[j] < A[minIndex] then
        minIndex =j
  swap (A[i],A[minIndex])
```

a) Pokaži pravilnost delovanja algoritma z izbiro uporabljajoč zančno invarianco

b) Analiziraj njegovo časovno zahtevnost.

a) Zančna invarianca: na začetku i-te iteracije velja, da je podpolje $A[1,\dots,i-1]$ urejeno in vsebuje $i-1$ najmanjših elementov v A.

Baza: $i=1$. Na začetku 1. iteracije velja, da je podpolje $A[1,\dots,0]$ urejeno in vsebuje 0 najmanjših elementov v A.

Ind. korak: $i \mapsto i+1$. Pokažimo, da na začetku $(i+1)$ iteracije velja, da je podpolje $A[1,\dots,i]$ urejeno in vsebuje prvih i najmanjših elementov v A.
Dovolj je pokazati, da je na začetku $(i+1)$ iteracije element $A[i]$ i-ti najmanjši v A. Poglejmo si i-to iteracijo. Notranja zanka poišče i-ti najmanjši element v A, ki je shranjen v $A[\text{minIndeks}]$. Klic swap$(A[i], A[\text{minIndex}])$ zagotovi, da je i-ti najmanjši element v $A[i]$.

Poglejmo: $i=n$. Po zančni invarianci je podpolje $A[1,\dots,n-1]$ urejeno in vsebuje $(n-1)$ prvih najmanjših elementov. Očitno mora biti A urejeno.

11.10.$\overline{24}$

Vaje 2

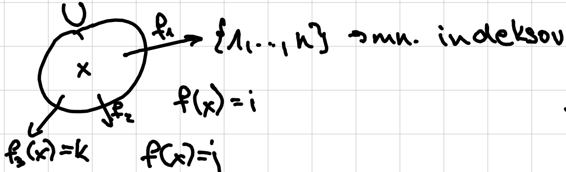
Programerske DN: navodila v pdf (DN 0 ni obvezna, priporočljiva)

$\downarrow$

Na učilnici

Bloomov Filter

$B = [\underset{1}{0}, \underset{\dots}{\dots}, \underset{n}{0}] \rightarrow$ polje samih ničel



$B = [\underset{1}{0}, \overset{i}{1}, \dots, \overset{j}{1}, \underset{n}{0}]$

$x \overset{?}{\in} B$

Vaje 1, nal. 3 do konca

```
for i=1 to n-1 do
  minIndex=i
  for j=i+1 to n
    if A[j] < A[minIndex] then
      minIndex =j
  swap (A[i],A[minIndex])
```

čas. zahtevnost

$c_1 \cdot n$

$c_2 (n-1)$

$c_3 \cdot \sum_{i=1}^{n-1} (n-i+1)$

$c_4 \cdot \sum_{i=1}^{n-1} (n-i)$

$c_5 \sum_{i=1}^{n-1} t_i$

$c_6 \cdot (n-1)$

t_i ... št., ki pove, kolikokrat je if stavek izpolnjen pri danem i.

Splošno:

$$T(n) = c_1 n + c_2(n-1) + c_3 \sum_{i=1}^{n-1}(n-i+1) + c_4 \sum_{i=1}^{n-1}(n-i) + c_5 \sum_{i=1}^{n-1} t_i + c_6(n-1)$$

Best case ($t_i = 0, \forall i \Leftrightarrow$ A je že urejen)

$$T(n) = c_1 n + c_2(n-1) + c_3 \sum_{i=1}^{n-1}(n-i+1) + c_4 \sum_{i=1}^{n-1}(n-i) + c_6(n-1)$$

- DN: Poračunaj vsote (podoben worst case-u)

Worst case ($t_i = n-i, \forall i \Leftrightarrow$ A je urejen v obratnem vstnem redu)

$$T(n) = c_1 n + c_2(n-1) + c_3 \sum_{i=1}^{n-1}(n-i+1) + c_3 \underbrace{\sum_{i=1}^{n-1} 1}_{=(n-1)} + c_4 \sum_{i=1}^{n-1}(n-i) + c_5 \sum_{i=1}^{n-1}(n-i)$$
$$+ c_6(n-1)$$

$$= (-c_2 - c_3 - c_6) + (c_1 + c_2 + c_3 + c_6) n + (c_3 + c_4 + c_5) \sum_{i=1}^{n-1}(n-1)$$

Vsota 1. m naravnih št.:

$$\sum_{i=1}^{m} i = 1+2+3+\dots+m = \frac{m(m+1)}{2}$$

$$= (-c_2 - c_3 - c_6) + (c_1 + c_2 + c_3 + c_6) n + (c_3 + c_4 + c_5) \frac{(n-1)\cdot n}{2}$$

$$= (-c_2 - c_3 - c_6) + (c_1 + c_2 + \frac{1}{2}c_3 - \frac{1}{2}c_4 - \frac{1}{2}c_5 - \frac{1}{2}c_6)n + \frac{(c_3 + c_4 + c_5)}{2} n^2$$
$$= A + Bn + Cn^2 = \Theta(n^2)$$

1. Imamo naslednjo kodo:

```
int FooBar(A)
    n = len(A); B = newarray(n)
    for j := 0 to n-1 {
        B[j] = 0
        for i := j to 0 step -1
            if A[i] < A[j] {
                B[j] = B[j] + 1
            }
    }
}
return B
```

a) Naj bo A[56, 47, 66, 71, 17, 19, 92]. Kaj vrne FooBar?

b) Kaj vsebuje vektor B, ki ga vrne FooBar(A) v splošnem?

c) Poišči zančno invarianco za notranjo for zanko in jo pokaži.

d) Analiziraj čas. zahtevnost alg. (DN)

c) = dokaz za a) in b)

a) $n = 7$

$$\begin{array}{cccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ B = [& 0, & 0, & \cancel{0}, & \cancel{0}, & 0, & \cancel{0}, & \cancel{0}] \\ & & & \cancel{1} & \cancel{1} & & 1 & 6 \\ & & & 2 & \cancel{2} & & & \\ & & & & 3 & & & \end{array}$$

$\Rightarrow B = [0, 0, 2, 3, 0, 1, 6]$

$j = 0$: $i = 0$ //

$j = 1$: $i = 1$ //
$i = 0$ //

$j = 2$: $i = 2$ //
$i = 1$ +1
$i = 0$ +1

$j = 3$: $i = 3$ //
$i = 2$ +1
$i = 1$ +1
$i = 0$ +1

b) B[j] ... št. ele. v A[0,...,j], ki je manjših od A[j]

c) Z.I. : Pri danem j, po vsaki iteraciji i velja, da je v $B[j]$ shranjeno št. ele. v $A[i, \ldots, j]$, ki so manjši od $A[j]$

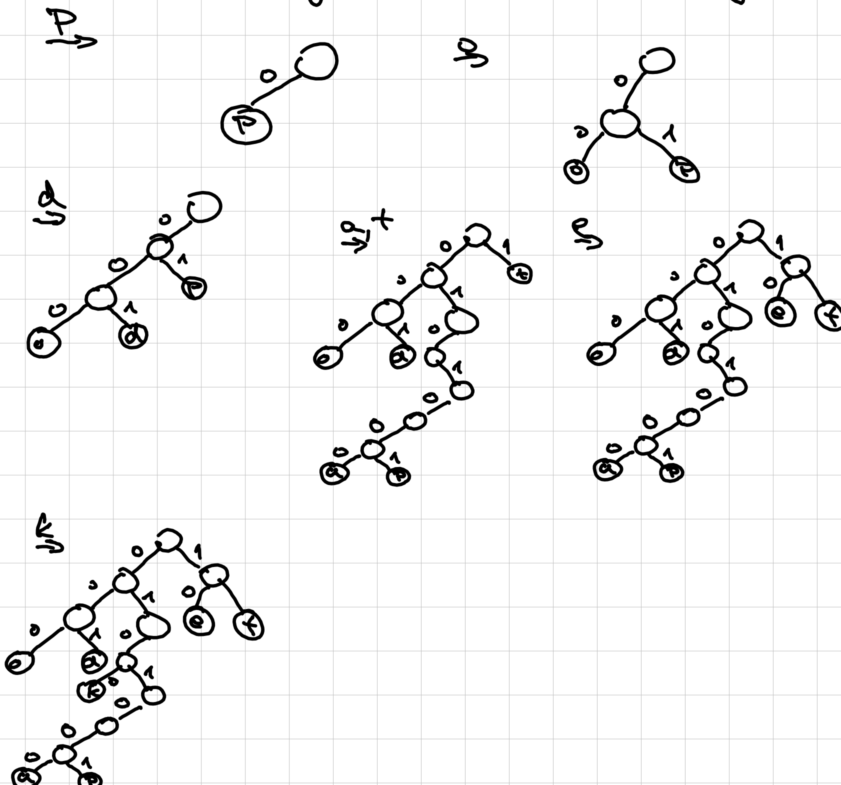
Dokaz z indukcijo:

BI: $i=j$: št. ele. v $A[j, \ldots, j]$ je 1 in št. manjših od $A[j]$ je 0, kar velja ker je $B[j]=0$ inicializiran

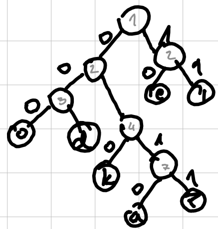
IK: $i \mapsto i-1$: Pokazati moramo, da po iteraciji $(i-1)$ velja, da je v $B[j]$ shranjeno št. ele. v $A[i-1, \ldots, j]$, ki so manjši od $A[j]$. Po I.P., po iteraciji i velja, da je v $B[j]$ shranjeno št. ele. v $A[i, \ldots, j]$, ki so manjši od $A[i]$.
Poglejmo si iteracijo $(i-1)$. V tej iteraciji primerjamo $A[i-1]$ z $A[j]$. Če je $A[i-1] < A[j]$, vrednosti $B[j]$ prištejemo 1 in rezultat sledi.

Vaje 3

1. Imamo dvojiško drevo $\Sigma = \{0,1\}$ in naslednje elemente:
(01010011, p), (00000111, o), (00100001, d), (01010001, a),
(11101100, t), (10010101, e), (01001010, k).
a) Vstavi zg. el. v številsko drevo.
b) Nariši pripadajoče Patricijevo drevo.
c) Patricijevo drevo stisni po prvih dveh plasteh, prvih treh platsteh, prvih štirih plasteh in prvih petih plasteh. Za katero stiskanje bi se odločili in zakaj?

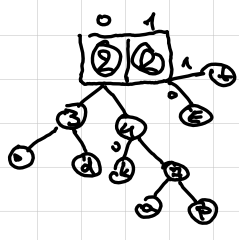
a) 

b)

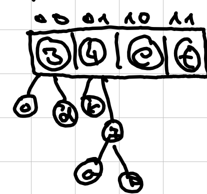


Čas. zahtevnost ostane enaka
Pros. zahtevnost se zmanjša

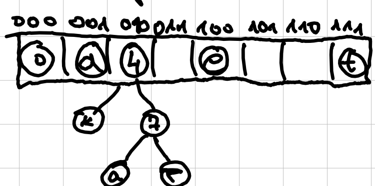
c) Po 2 plasteh



Po 3 plasteh



Po 4 plasteh



Po 5 plasteh (DN

7. Imamo naslednje ključe nad abecedo $\Sigma = \{0,1\}$:

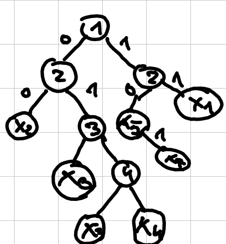
$x_1 = 111110$
$x_2 = 0001010110$
$x_3 = 0110000$
$x_4 = 011110$
$x_5 = 101$
$x_6 = 01011$
$x_7 = 1011111$

a) Vstavi dane ključe v Patricijevo drevo.

b) Patricijevo drevo stisnite po plasteh za $\alpha = \frac{3}{4}$

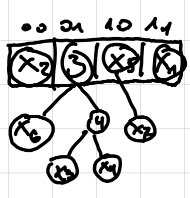
c) Recimo, da imamo v št. drevesu n ključev. Kakšne naj bodo njihove vrednosti, da po stiskanju po plasteh $\alpha = \frac{3}{4}$ ne bo nobenega polja? D.r., za vsako stiskanje po plasteh bi bila gostota vedno prenizka

a)
$x_2 = 0001010110$
$x_6 = 01011$
$x_3 = 0110000$
$x_4 = 011110$
$x_5 = 101$
$x_7 = 1011111$
$x_1 = 111110$



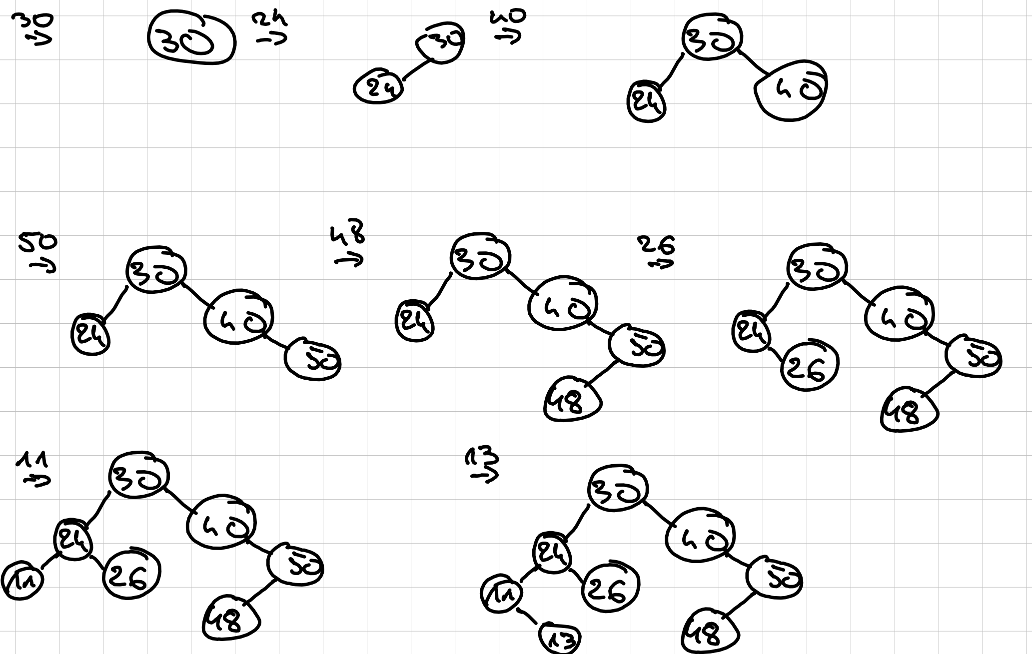
b)

1. $1/1 = 1 > \frac{3}{4}$
2. $2/2 = 1 > \frac{3}{4}$
3. $4/4 = 1 > \frac{3}{4}$
4. $5/8 < \frac{3}{4}$



Vaje 4

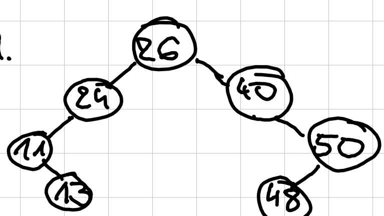
1. Ustavi naslednje elemente v dvojiško iskalno drevo: 30, 24, 40, 58, 48, 26, 11, 13



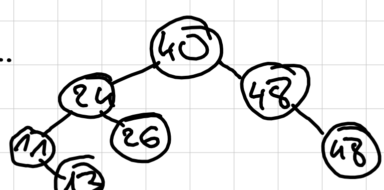
Izbriši 30 iz zadnjega drevesa.

1. Zamenjamo ga lahko z največjim el. v levem poddrevesu
2. Zamenjamo ga lahko z najmanjšim el. v desnem poddrevesu

1.



2.



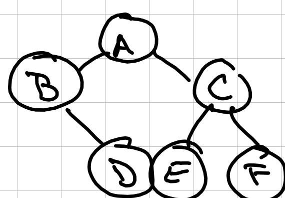
Naredi premi, umesni in obratni pregled drevesa.

Premi (KLD): 30, 24, 11, 13, 26, 40, 58, 48

Umesni (LKD): 11, 13, 24, 26, 30, 40, 48, 58

Obratni (LDK): 13, 11, 26, 24, 48, 58, 40, 30

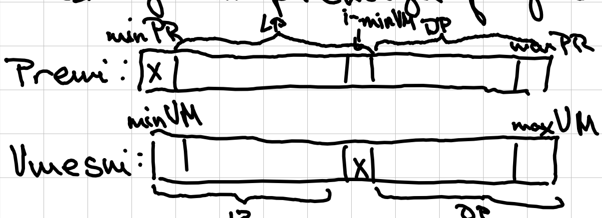
2. Dano je naslednje iskalno drevo:



a) Poišči umesni pregled drevesa:

B, D, A, E, C, F

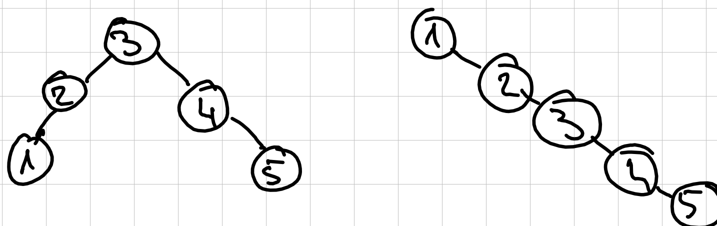
b) Opiši alg., ki zgradi dvojiško drevo iz danega umesnega in premega pregleda.



1. Za koren vzemi prvi element v premem pregledu
   x = Premi[minPR]
2. Poišči i, da Umesni[i] = x
3. Rekurzivno zgradi levo poddrevo na
   Premi[minPR+1, ..., i - minVM]
   Umesni[minVM, ..., i - 1]
   in desno poddrevo na

Premi[i-minVM+1, ..., maxPR]
Vmesni[i+1, ..., maxVM]

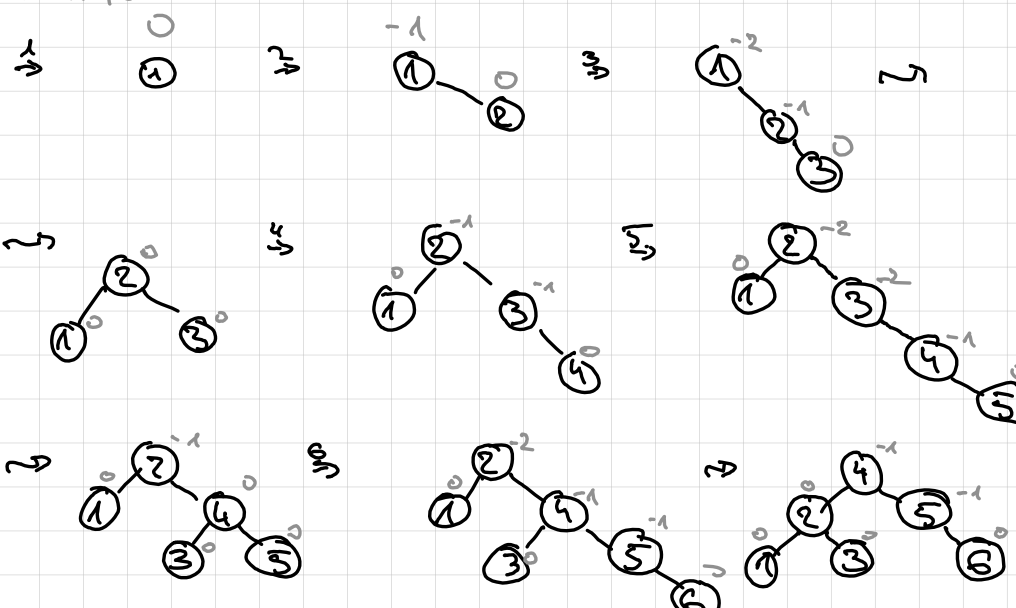
c) Pokaži da vmesni pregled ne zadošča, da lahko rekonstruiramo drevo



Vmesni: 1, 2, 3, 4, 5 1, 2, 3, 4, 5

3. Vstavi v AVL drevo zaporedna št. 1-7.

Uravnoteženost (L-D):



Če je $n = 2^k - 1$ za nek k, kako izgleda AVL drevo z elementi od 1 do n.

Vaje 5 8.11.24

1. Uporabi dvojiška drevesa za urejanje n št. in opiši ter analiziraj časovno zahtevnost takšnega algoritma.

Koraki:

→ AVL je boljši način

1) Zgradi (dvojiško iskalno drevo) iz teh n elementov ⇓ $O(n \log n)$
2) Naredi vmesni pregled na zgrajenem drevesu ⇓ $O(n)$

$\log(1)$
$\log(2)$
$\vdots$
$\log(n) = O(\log(n))$

$\Sigma \Rightarrow O(n \log n)$

$\left.\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ \vdots \\ n \end{array}\right\} \Sigma \frac{n(n+1)}{2} \Rightarrow O(n^2)$

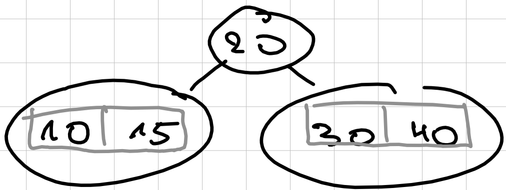
2. Vstavi naslednje elemente v B-drevo reda $b=5$:
20, 40, 10, 30, 15, 35, 7, 26, 18, 22, 5, 42, 13, 46, 27, 8, 32, 24, 45, 25

→ Najlažje taka vozlišča predstavimo s povezanim seznamom

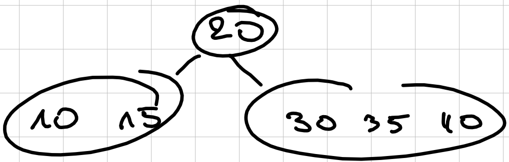
$\xrightarrow{20}$ 20 $\xrightarrow{40}$ | 20 | 40 | $\xrightarrow{10}$ | 10 | 20 | 40 |

$\xrightarrow{30}$ | 10 | 20 | 30 | 40 | $\xrightarrow{15}$ | 10 | 15 | 20 | 30 | 40 |

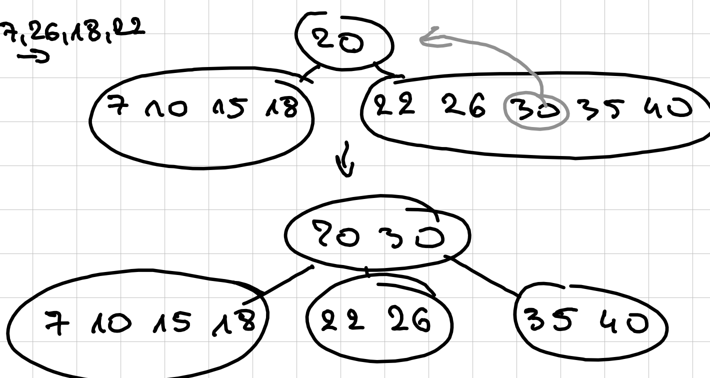
15 →



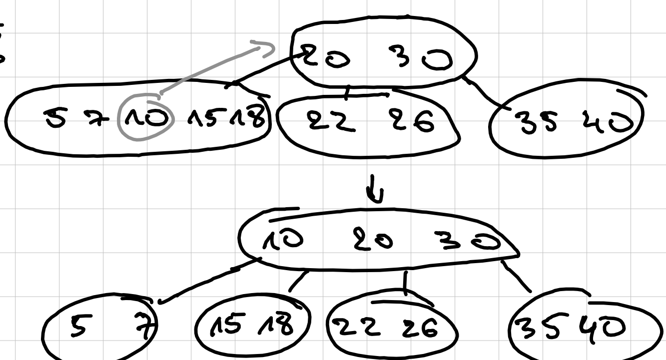
35 →



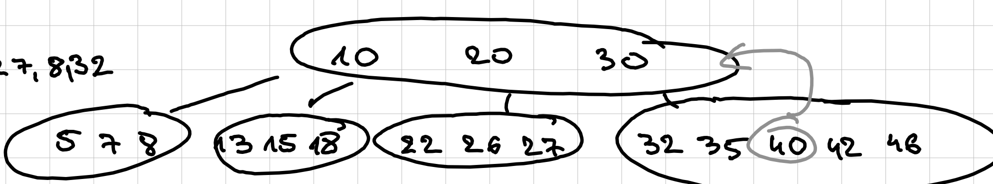
7, 26, 18, 22 →

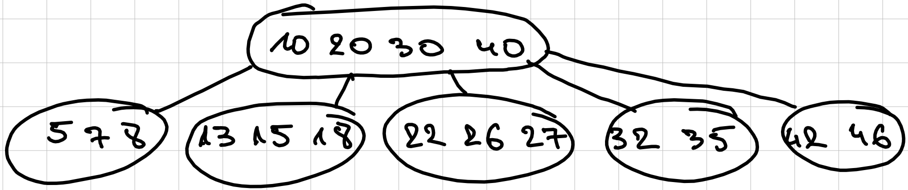


5 →

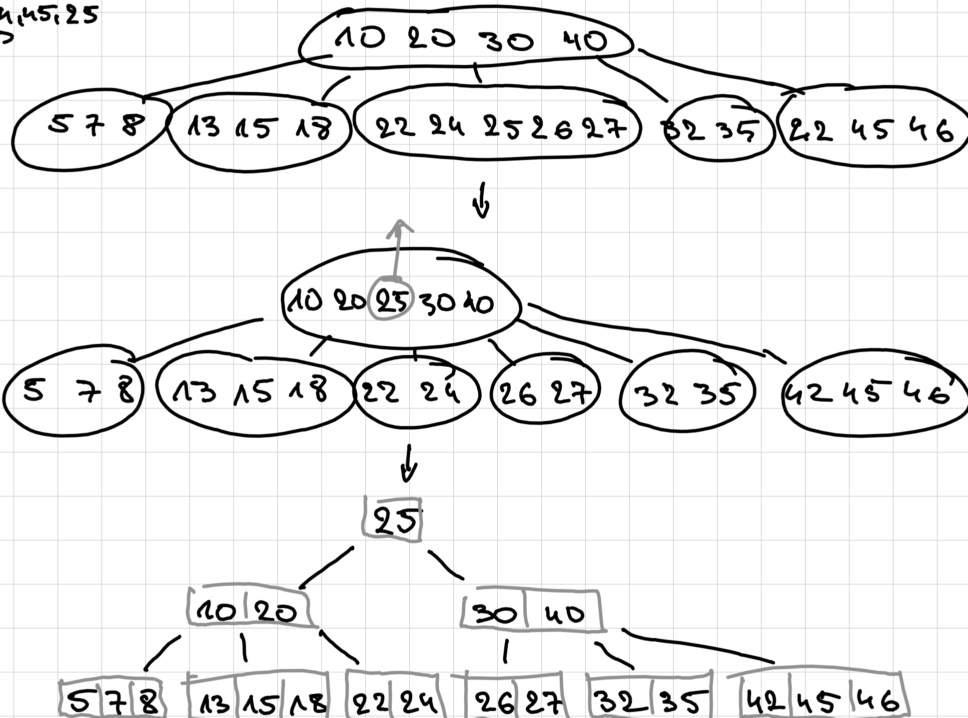


42, 13, 46, 27, 8, 32 →



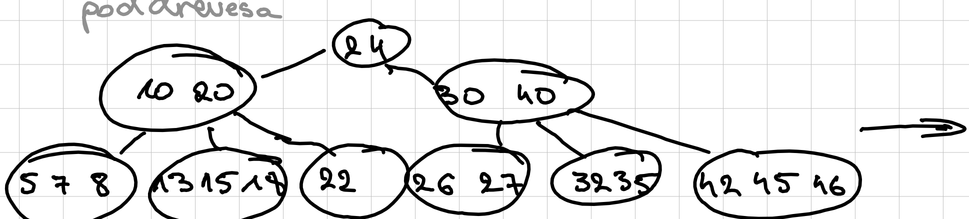


24, 45, 25
→

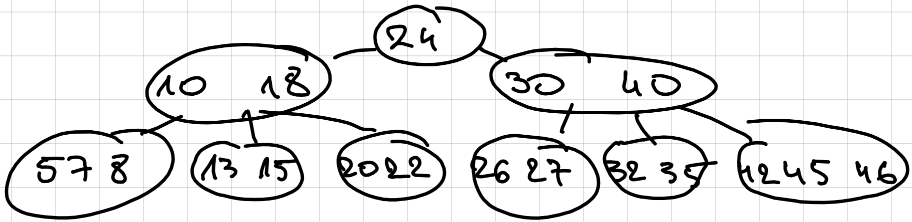


3. Iz zadnjega B drevesa izbriši 25, 45, 24, 32.

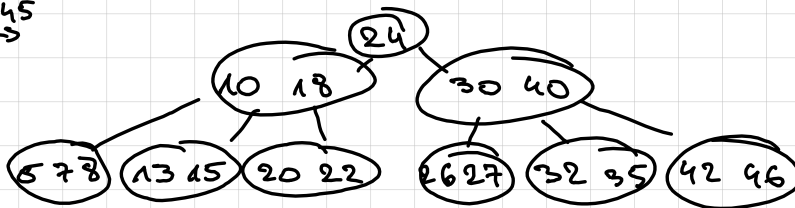
1) -25 za novo vozlišče določimo najmanjši el. iz desnega poddrevesa ali največji el. iz levega poddrevesa



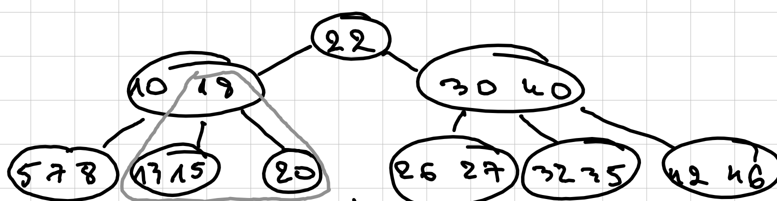
če je v listu le 1 el. potem si sposodimo, ga nesemo nivo višje in max el. iz korena nesemo v pravi list.

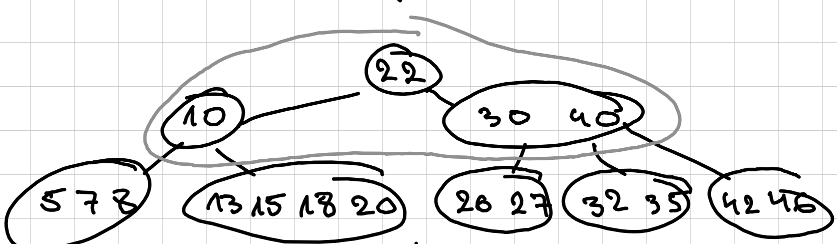

24
10 18
30 40
5 7 8
13 15
20 22
26 27
32 35
42 45 46

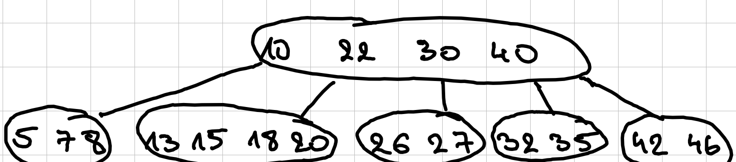
-45


24
10 18
30 40
5 7 8
13 15
20 22
26 27
32 35
42 46

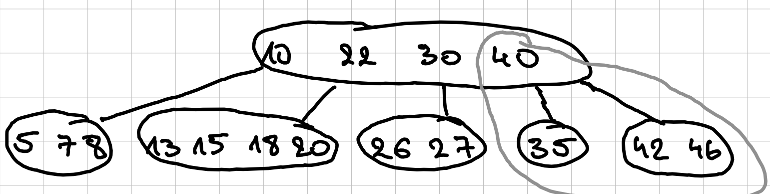
-24

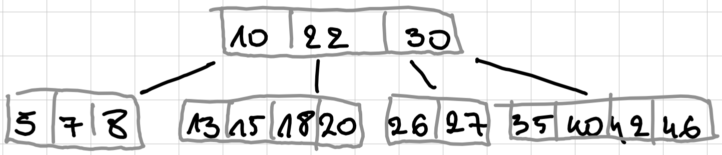

22
10 18
30 40
5 7 8
13 15
20
26 27
32 35
42 46


22
10
30 40
5 7 8
13 15 18 20
26 27
32 35
42 46

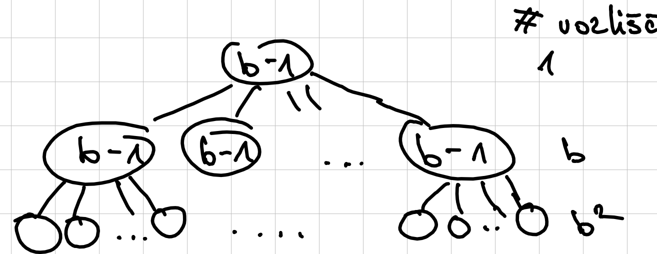

10 22 30 40
5 7 8
13 15 18 20
26 27
32 35
42 46

-32


10 22 30 40
5 7 8
13 15 18 20
26 27
35
42 46



4. Največ koliko elementov ima lahko B-drevo reda b z višino h.



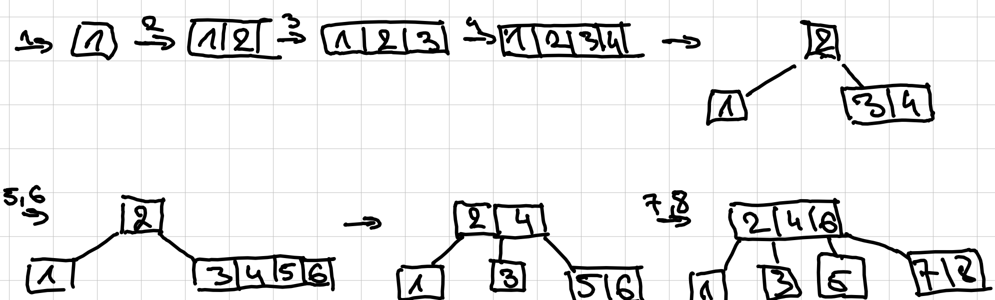
Višina:	# elementov
1	$(b-1)$
2	$b(b-1)$
3	$b^2(b-1)$
$\vdots$	$\vdots$
i	$b^{i-1}(b-1)$
$\vdots$	$\vdots$
h	$b^{h-1}(b-1)$

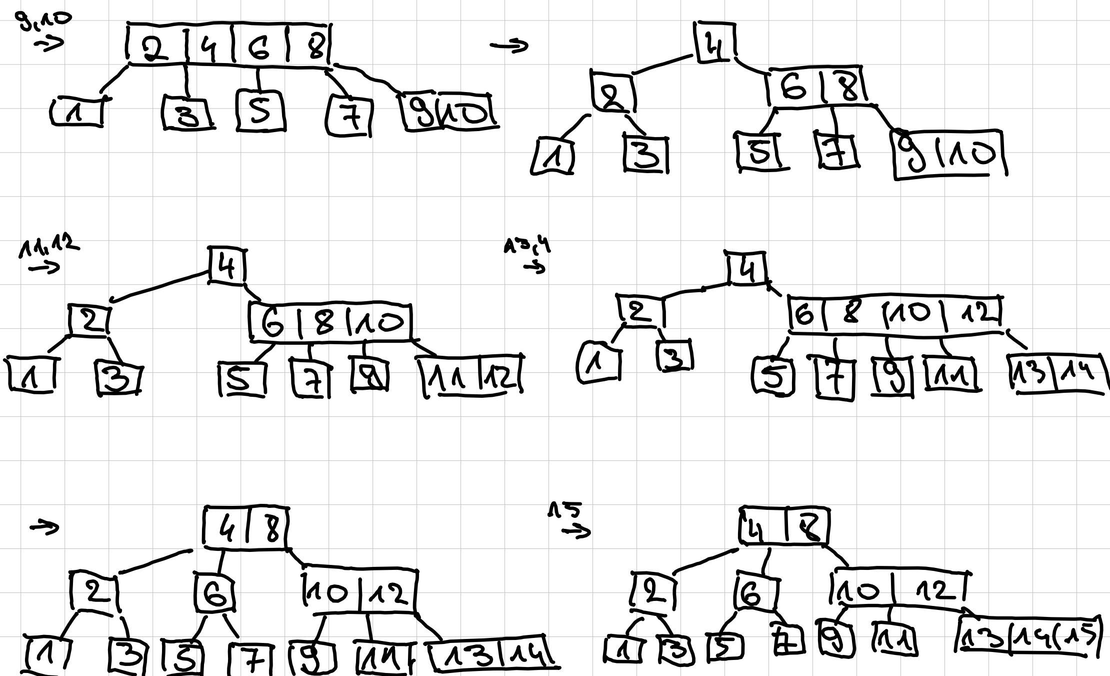
max # el.

$$\sum_{i=1}^{h} b^{i-1}(b-1) = (b-1)\cdot\sum_{i=1}^{h} b^{i-1}$$

$$= (b-1)\,\frac{b^h-1}{b-1} = b^h-1$$

5. Vstavi števila od 1 do 15 v drevo reda b=4. Recimo, da vstavimo št. od 1 do n v drevo reda b=4. Opiši, kako izgleda takšno drevo. Kakšna je njegova višina



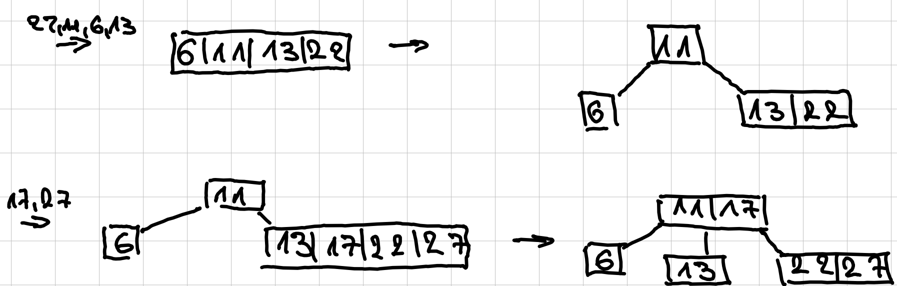


Vsak najmanjši element v korenu lahko izračunamo po formuli $2^{(\log n)}$

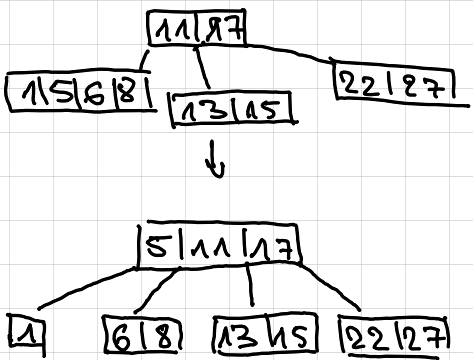
Vaje 6

15.11.24

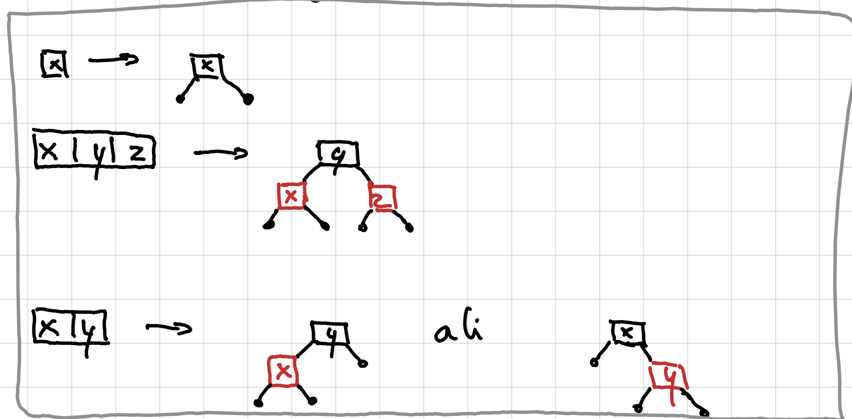
1. Vstavi naslednje elemente v B-drevo reda b=4:
   22, 11, 6, 13, 17, 27, 8, 15, 5, 1

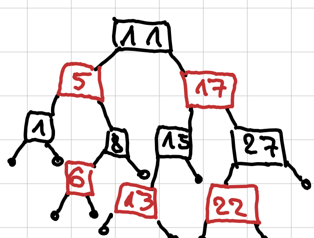


8, 15, 5, 1 →



2. Pretvori zadnje B-drevo u rdeče-črno drevo





3. Vzporedno vstavi naslednje elemente v rdeče-črno drevo in pripadajoče B-drevo, b=4:
4, 7, 12, 15, 3, 5, 14, 18, 16, 17

B-drevo

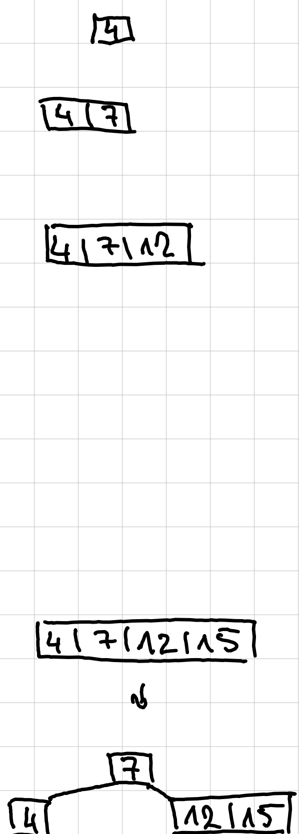
R-Č drevo

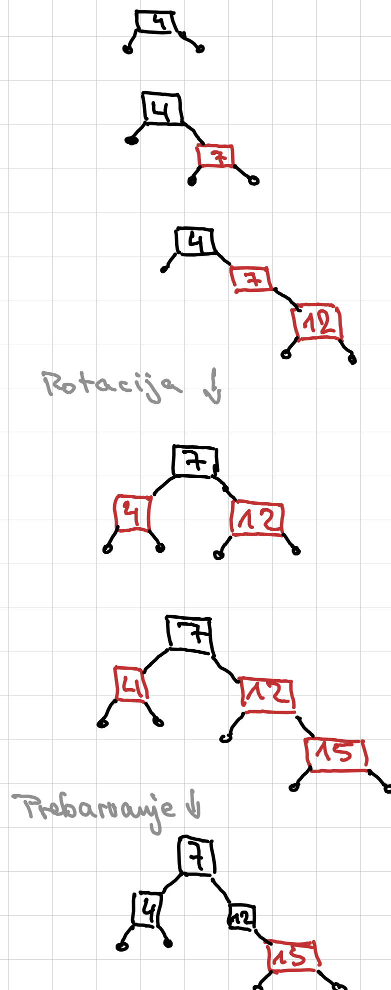
4 →

7 →

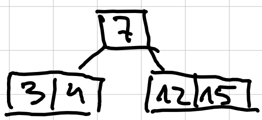
12 →

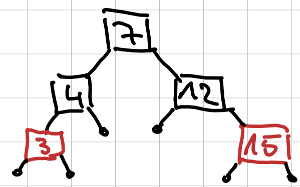
15 →



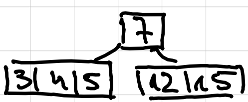


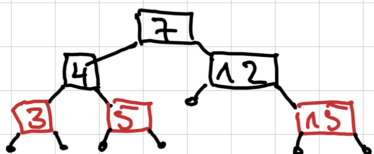
3 →



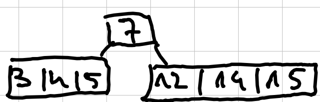


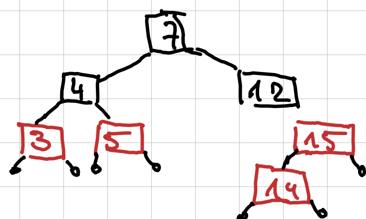
5 →



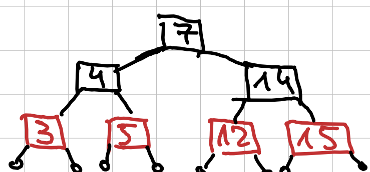


14 →

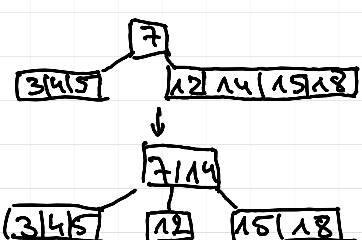


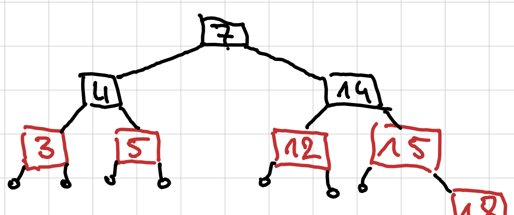


Rotacija ↓

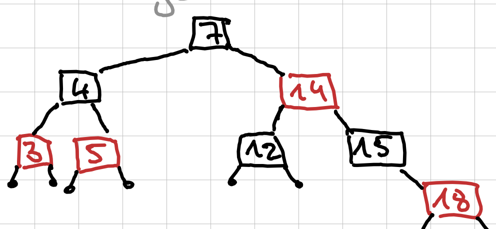


18 →

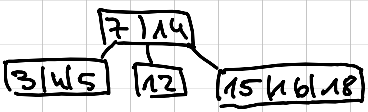


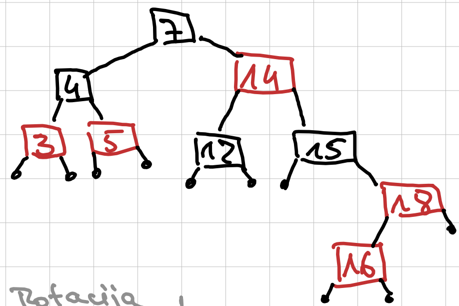


Prebarvanje ↓

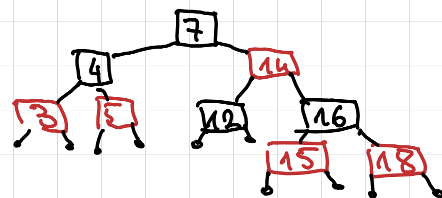


16 →

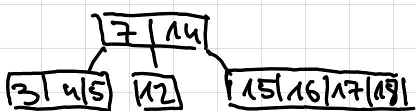




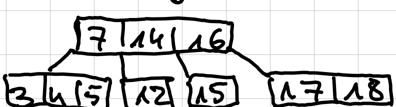
Rotacija ↓

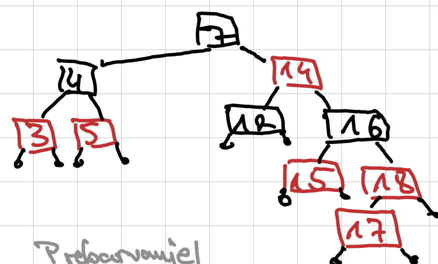


17 →

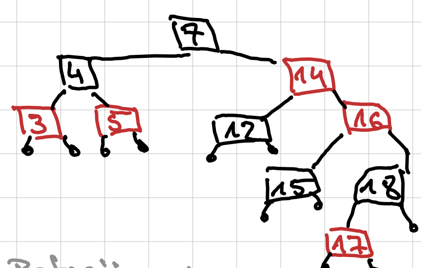


↓

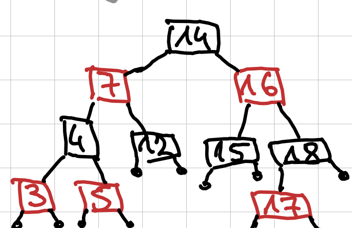




Prebarvanje ↓



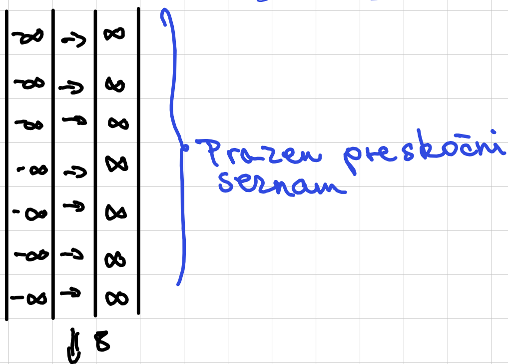
Rotacija ↓

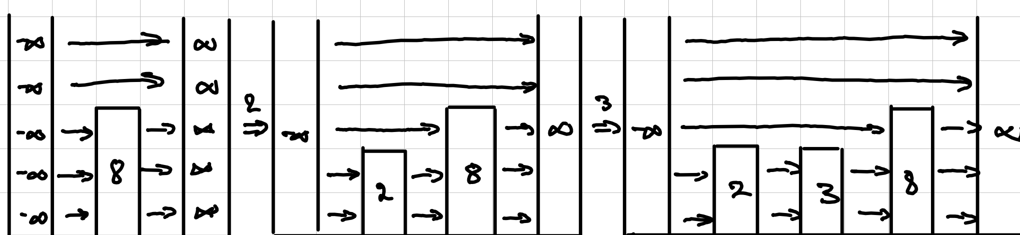


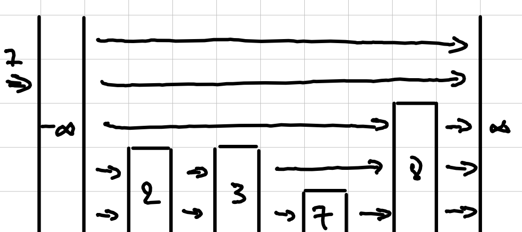
4. V preskočni seznam vstavi naslednje elemente: 8, 2, 3, 7, 13, 19, 12, 26; kjer generator naključnih št. vrne naslednje zaporedje:

1, 1, 0 | 1, 0 | 1, 1, 0 | 0, | 0, | 0, | 1, 1, 1, 0, | 1, 0 | 0 0, 1, ...

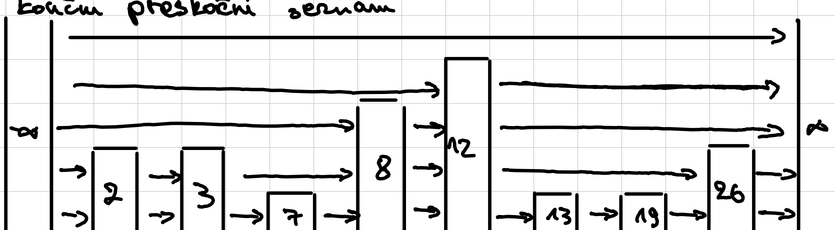
8 2 3 7 13 19 12 26



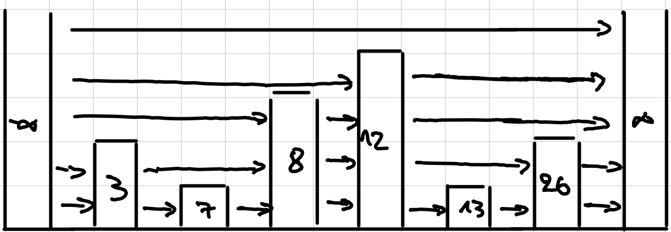




končni preskočni seznam

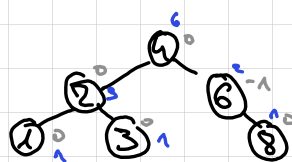


Izbriši 13 in 2 iz končnega pres. seznama.



Vaje 7

22.11.24



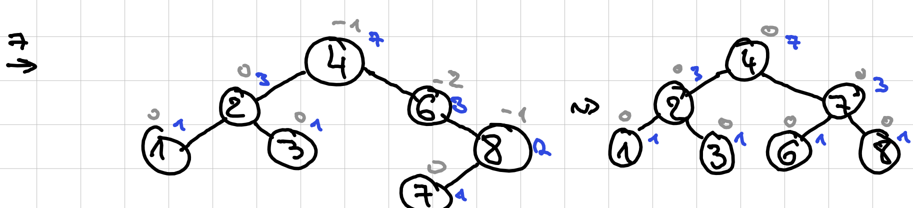
Rang(3) = 3

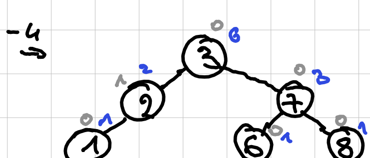
$T_{RANG} = O(\log n)$

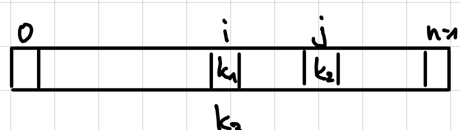
: Dodatna info

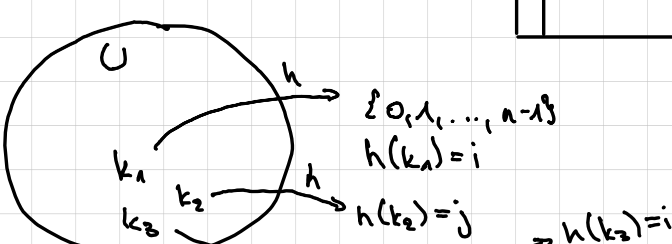
Z razširjanjem strukture se informacije spremenijo

Vstavi 7:









1. Vstavi naslednje ključe: 10, 22, 31, 4, 15, 28, 17, 88, 59 v razpršilno tabelo velikosti $n=5$ z uporabo veriženja in razpršilno f-jo $h(k) = 13 \cdot k \bmod n$.

0	1	2	3	4
↓	↓	↓	↓	↓
10	22	4	31	28
↓	↓	↓		↓
15	17	59		88

$h(10) = 13 \cdot 10 \bmod 5 \equiv 0$
$h(22) = 13 \cdot 22 \bmod 5 \equiv 1$
$h(31) = 13 \cdot 31 \bmod 5 = 3$
$h(4) = 13 \cdot 4 \bmod 5 \equiv 2$
$h(15) = 13 \cdot 15 \bmod 5 \equiv 0$
$h(28) = 13 \cdot 28 \bmod 5 \equiv 4$
$h(17) = 13 \cdot 17 \bmod 5 \equiv 1$
$h(88) = 13 \cdot 88 \bmod 5 \equiv 4$
$h(59) = 13 \cdot 59 \bmod 5 \equiv 2$

Iste elemente vstavi še v razpršilno tabelo velikosti $n=11$ z uporabo odprtega naslavljanja in razpršilno f-jo $h'(k) = k$:

a) linearno: $h(k,i) = (h'(k) + i) \bmod n$
b) kvadratično: $h(k,i) = (h'(k) + i^2 + 3i) \bmod n$
c) dvojno: $h(k,i) = (h'(k) + i \cdot h''(k)) \bmod n$,
   kjer $h''(k) = 1 + (k \bmod (n-1))$

a)

isDeleted	0	0	0	0	1	0	0	0	0	0	0
	0	1	2	3	4	5	6	7	8	9	10
	22	88			~~4~~	15	28	17	59	31	10

$h(10,0) = (10+0) \bmod 11 \equiv 10 \bmod 11$
$h(22,0) = (22+0) \bmod 11 \equiv 0 \bmod 11$
$h(31,0) = 31 \bmod 11 \equiv 9 \bmod 11$
$h(4,0) = 4 \bmod 11 \equiv 4 \bmod 11$
$h(15,0) = 15 \bmod 11 \equiv 4 \bmod 11$
$h(15,1) = 16 \bmod 11 \equiv 5 \bmod 11$
$h(28,0) = 28 \bmod 11 \equiv 6 \bmod 11$
$h(17,0) = 17 \bmod 11 \equiv 6 \bmod 11$
$h(17,1) = 18 \bmod 11 = 7 \bmod 11$

$h(88,0) = 88 \bmod 11 \equiv 0 \bmod 11$
$h(88,1) = 89 \bmod 11 \equiv 1 \bmod 11$
$h(59,0) = 59 \bmod 11 \equiv 4 \bmod 11$
$h(59,1) \equiv 5 \bmod 11$
$h(59,2) \equiv 6 \bmod 11$
$h(59,3) \equiv 7 \bmod 11$
$h(59,4) \equiv 8 \bmod 11$

2. Peter je našel fjo Zmešji(k), ki vzame celo število k in vrne vrednost na int. $[0, m)$. Poleg tega ve, da je Zmešji(k) skoraj popolna razpr. fja. $(\forall k)(\forall j)\ P([\text{Zmešji}(k)] = j) = \frac{1}{m}$
a) Peter želi uporabiti Zmešji (k) za implementacijo slovarja na polju $a[0, \ldots, n-1]$. Predpostavi, da $n = m$. Problem je soupadanje. Napiši fji za vstavljanje in iskanje z uporabo Zmešji(k) in lin. naslavljanja.
b) Peter se odloči, da bo zamenjal n $(n < m)$. Ali lahko še vedno učinkovito uporabi Zmešji (k)?

a)

$$\left.\begin{array}{l} h: \mathbb{Z} \to \{0, \ldots, n-1\} \\ \text{Zmešji}: \mathbb{Z} \to [0, m) \end{array}\right\} n = m$$

$$h'(k) = [\text{Zmešji}(k)]$$

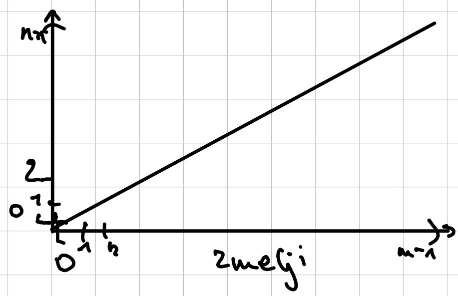
$$\text{lin.}: h(k, i) = ([\text{Zmešji}(k)] + i) \bmod n$$

```
Vstavi k:
   for i = 0,...,n-1:
      if a[h(k,i)] == NIL then
         a[h(k,i)] := k
         return true
   return false

Išči k:
   for i = 0,...,n-1:
      if a[h(k,i)] == k then
         return true
      else a[h(k,i)] == NIL and IsDeleted == 0 then
         return false
   return false
```

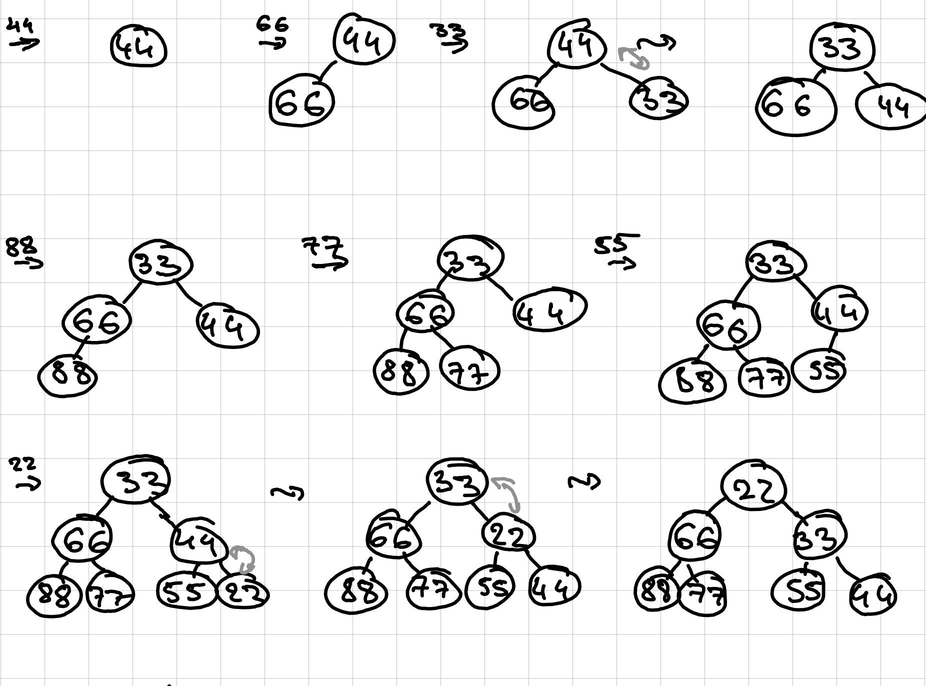
b)

$h'(k) = \left[\frac{n}{m} \cdot \text{Zmelji}(k) \right]$



Vaje 8 — 29.11.24

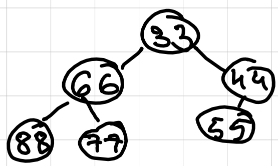
1. Vstavi naslednje elemente v binarno kopico:
44, 66, 33, 88, 77, 55, 22



$T(n) = O(\log n)$ → v najslabšem primeru moramo zamenjati koren z listom ($h = \log n$)

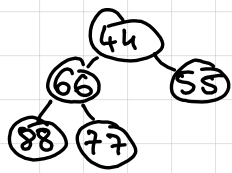
1. rešitev za 2x DeleteMin()

DeleteMin()
→



Izberemo manjšega od otrok in ga damo za nov koren. Rekurz. ponovimo po celem podd.

DeleteMin()
→



2. rešitev za 2x DeleteMin()

DeleteMin()
→

DeleteMin()
→

$T_{Min}(n) = O(1)$

$T_{DelMin}(n) = O(\log n)$

2. Opiši podatkovno strukturo, ki podpira naslednje operacije: Vstavi(x), IzbrišiZadnjega() - izbriše el., ki največ časa v strukturi - in PoiščiMin(). Prvi dve operaciji naj imata $O(\log n)$ čas. zah., zadnja pa $O(1)$.

Op. vrste s prednostjo

Op. vrste

2 PS z istimi elementi

→ Kopica

→ Vrsta - implementiranje kazalcev na kopico, rep, glavo (povezan seznam)

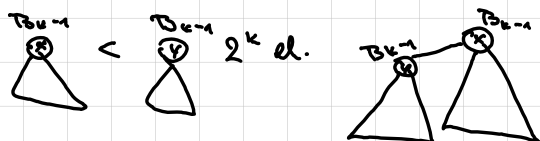
PoiščiMin() išče le po kopici
Vstavi(x) vstavi v kopico, ker v PS kazalec na rep, v konst. času vstavi v PS
IzbrišiZadnjega() izbriše iz kopice in postavimo na njegovo mesto skrajno desnega iz zadnjega nivoja. Po potrebi potopimo oz. dvignemo element. Iz PS brišemo v konst. času.

Binomsko drevo B_k reda k

B_0: 1 el

B_1: 2 el B_2: 4 el

B_k: 2^k el.



Binomsko drevo B_k vsebuje 2^k elementov.

3. Pokaži, da ima drevo B_k na i-tem nivoju $\binom{k}{i}$ elementov za fiksen i.

$$\binom{k}{i} = \frac{k!}{i!(k-i)!}$$

Indukcija po k:
$k=0$: B_0 na item nivoju $\binom{0}{i} = 1$ element

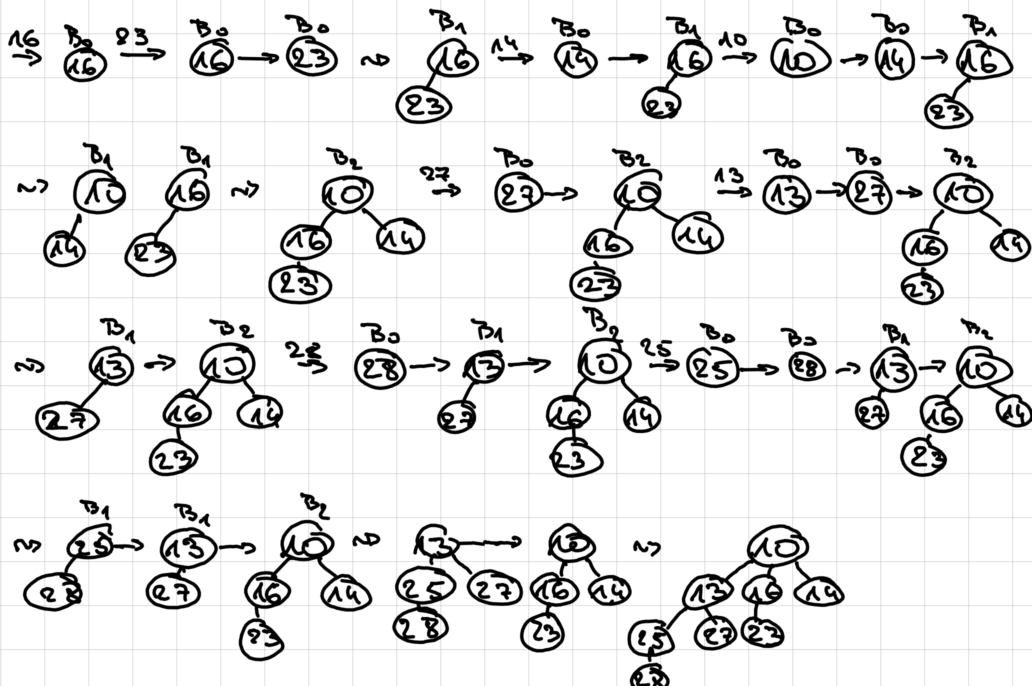
$k \to k+1$: Recimo, da je formula res za drevesa reda k.
Pokažimo, da ima drevo B_{k+1} na item nivoju $\binom{k+1}{i}$ elementov
št. el. na i-tem nivoju v B_{k+1} = št. el v B_k na i nivoju + št. el. v B_k na i-1 nivoju

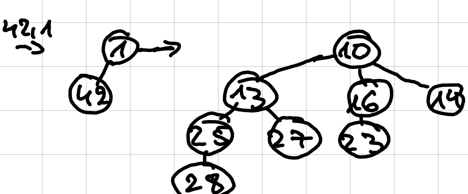
Št. el. B_{k+1} na i nivoju $= \binom{k}{i-1} + \binom{k}{i}$

$$= \frac{k!}{(i-1)!(k-i+1)!} + \frac{k!}{i!(k-i)!} = \frac{k!\cdot i + k!(k-i+1)}{i!(k-i+1)!}$$

$$= \frac{k!(k+1)}{i!((k+1)-i)!} = \frac{(k+1)!}{i!((k+1)-i)!} = \binom{k+1}{i}$$

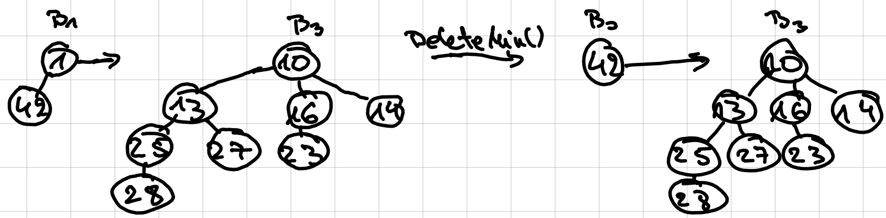
4. Vstavi naslednje elemente v binomsko kopico:
16, 23, 14, 10, 27, 13, 28, 25, 42, 1

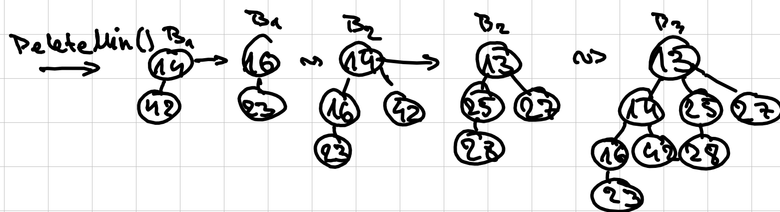


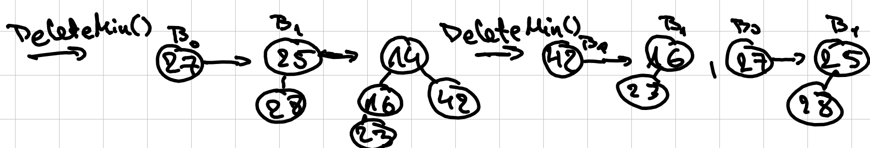


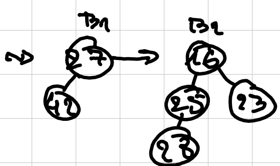
$2^{k+1} = n+1$
$k+1 = \log(n+1)$
$k = \log(n+1) - 1$
$k = O(\log n)$

Čas. zah.: $B_0, \dots, B_k$
$1 \; \dots \; 2^k$ elem. $\Rightarrow 2^{k+1} - 1$ el. $= n$

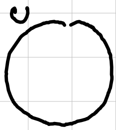








Vaje 8 6.12.24

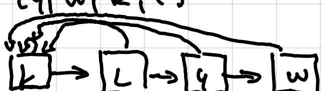


Preprosta unija s povezanim seznamom

MakeSet(x) $\{x\}$ $\{u,v,z\}$

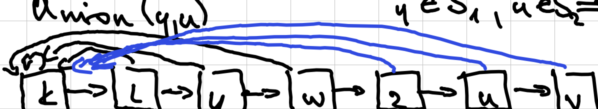
$O(1)$

$\{y,w,k,l\}$



FindRep(v) = z $\Rightarrow O(1)$

Union(y,u) $y \in S_1, u \in S_2 \Rightarrow O(|S_2|)$ Preprosta unija



1. Imamo preprosto implementacijo s povezanim seznamom.

a) Poišči zap. $n-1$ operacij Union na mn. $\{1\}, \{2\}, \dots, \{n\}$, ki zahteva $\Theta(n^2)$ časa.

Union(1,2)	1 popravek	$\{1,2\}, \{3\}, \dots \{n\}$
Union(3,1)	2 popravka	$\{1,2,3\}, \{4\}, \dots, \{n\}$
Union(4,1)	3 popravki	$\{1,2,3,4\}, \{5\}, \dots \{n\}$
$\vdots$	$\vdots$	$\vdots$
Union(n,1)	$n-1$ popravkov	$\{1,2,\dots n\}$
	$\Sigma = \frac{(n-1)n}{2} = \Theta(n^2)$	

Utežena unija: predstavnik unije je predstavnik tiste množice, ki je večja (če sta mn. enake velikosti, vzamemo predstavnika 1. mn.)

b) Pokaži da katerokoli zap. $n-1$ uteženih unij na mn. $\{1\}, \{2\}, \dots, \{n\}$ zahteva $O(n \log n)$ časa.

Poglejmo si poljuben $x \in \{1, \dots, n\}$ in preštejmo, kolikokrat se mu lahko največ spremeni referenca na predstavnika.

$$\underbrace{\text{Union}(x, x_1) \quad \underline{\underline{1}} \quad \begin{matrix} x \in S_1 \Rightarrow |S_1| = 1 \\ x_1 \in S_1' \Rightarrow |S_1'| \geq 1 \end{matrix}}$$

$$\underbrace{\text{Union}(x, x_2) \quad \underline{\underline{1}} \quad \begin{matrix} x \in S_2 = S_1 \cup S_1' \Rightarrow |S_2| \geq 2 \\ x_2 \in S_2' \Rightarrow |S_2'| \geq 2 \end{matrix}}$$

$$\underbrace{\text{Union}(x, x_3) \quad \underline{\underline{1}} \quad \begin{matrix} x \in S_3 = S_2 \cup S_2' \Rightarrow |S_3| \geq 4 \\ x_3 \in S_3' \Rightarrow |S_3'| \geq 4 \end{matrix}}$$

$\vdots$

$$\text{Union}(x, x_k) \quad \underline{\underline{1}} \quad \begin{matrix} x \in S_k = S_{k-1} \cup S_{k-1}' \Rightarrow |S_k| \geq 2^{k-1} \\ x_k \in S_k' \Rightarrow |S_k'| \geq 2^{k-1} \end{matrix}$$

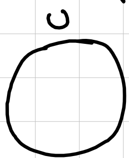
$O(n \log n)$

To naredimo n-krat

$2^{k-1} \leq |S_k| \leq n \Rightarrow 2^{k-1} \leq n \;/\log$

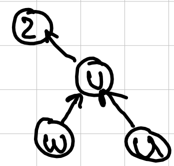
$k-1 \leq \log n \Rightarrow k \leq \log n \Rightarrow k = O(\log n)$

Preprosta unija z drevesa

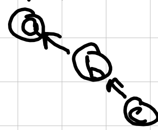


MakeSet(x) $\{x\}$

$\{u,v,z,w\}$

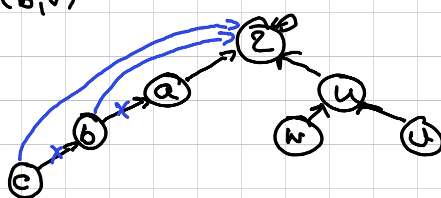


$\{a,b,c\}$



FindRep(w) = z

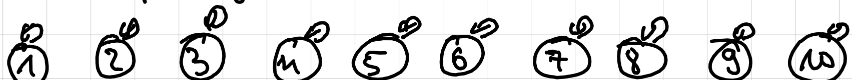
Union (b,v)



Unija je odvisna od globine drevesa

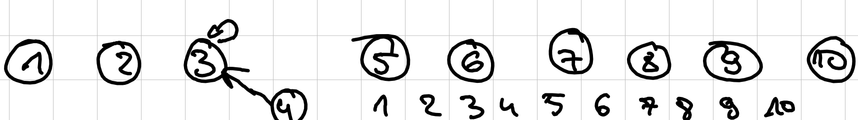
FindRep (c) = z

2. Implementacija z drevesi, utežena unija in stiskanje po poteh. Na mn. $\{1\},\dots,\{10\}$ naredimo naslednje op.: Union(3,4), Union(4,9), Union(8,10), Union(2,3), Union(5,6), Union(5,9), Union(7,3), Union(4,8), Union(6,1). Nariši PS disju. mn. po vsaki operaciji:



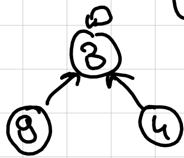
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10

Union(3,4)



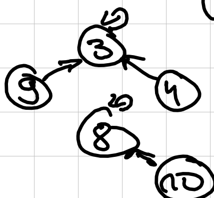
1	2	3	4	5	6	7	8	9	10
1	2	3	3	5	6	7	8	9	10

Union(3,9)



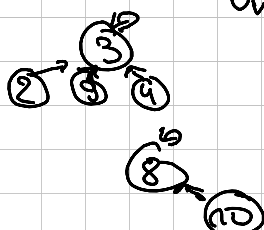
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	3	10

Union(8,10)



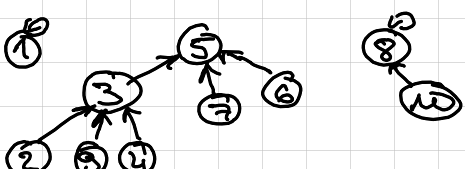
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	3	8

Union (2,3)



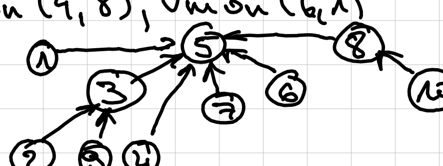
1	2	3	4	5	6	7	8	9	10
1	3	3	4	5	6	7	8	3	8

Union(5,6) Union (5,9), Union(7,3)



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

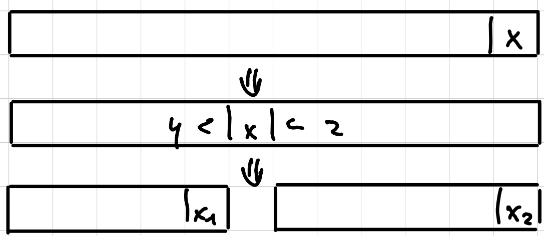
Union (4,8), Union (6,1)



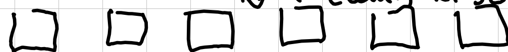
1	2	3	4	5	6	7	8	9	10
5	3	5	5	5	5	5	5	3	8

SORTIRANJE:

- Deli in vladaj (QuickSort)



Delimo in urejamo dokler nimamo skupkov ≤2 elem., ki so urejeni. Nato zlepimo



Koliko časa vzame QuickSort s pivotom kot zadnjim elementom na že urejenem polju? $O(n^2)$

Koliko časa vzame MergeSort na že urejenem polju?
$O(n \log n)$

Vaje 10 — 13.12.24

Dan je urejen seznam z n elementi. Dodamo $f(n)$ elementov v seznam. Kako bi uredili celoten seznam, če:

a) $f(n) = O(1)$? Vsak el. posebej vstavi na pravo mesto (kot pri urejanju z vstavljanjem) $O(n)$

b) $f(n) = O(\log n)$? Uredimo posebej drugi seznam:

$N = O(\log n) \Rightarrow O(N \log N) =$

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

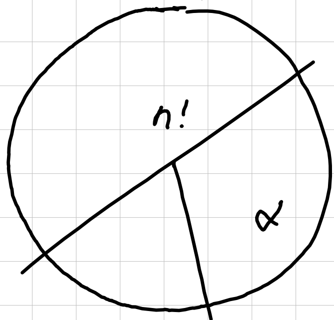
$= O(\log^2 n) = O(n)$

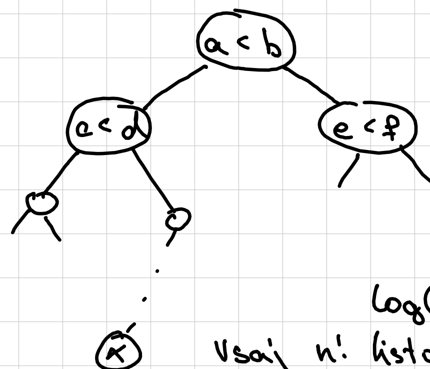
Zlijemo oba seznama: $O(n + \log n) = O(n)$

Ne obstaja noben alg., ki bi imel manj od $O(n \log n)$ primerjav.

n elementov $\Rightarrow$ problem urejanja

Prostor vseh možnih rešitev





$\log(n!) \approx n \log n$

Vsaj $n!$ listov

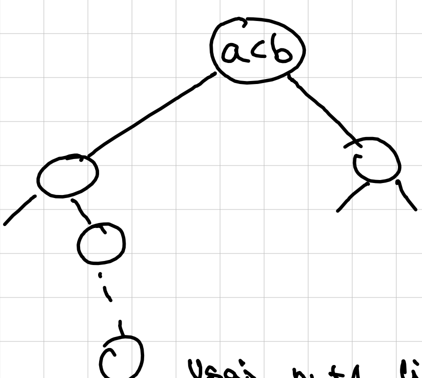
$h = \Omega(\log(n!))$

Dvojiško drevo z N listi. Koliko je najmanjša višina takega drevesa?

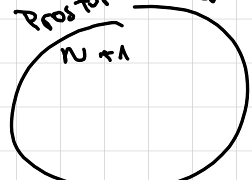
Polno drevo

$h = \log(2N-1)$

Dano je urejeno polje z n elementi. Vsaj koliko primerjav mora narediti katerikoli alg., da najde dani x v polju?



Prostor rešitev



Vsaj $n+1$ listov $h = \Omega(\log n)$

Zapiši $O(n)$-alg., ki uredi n celih št. z intervala $[0, n^3-1]$.

Korensko urejanje

$$\begin{array}{cccc} 5&4&3&\boxed{2}\\ 3&2&5&\boxed{1}\\ 4&1&6&\boxed{2}\\ 2&2&3&\boxed{6} \end{array}$$

Stabilno urejanje

$$\begin{array}{cccc} 3&2&5&1\\ 5&4&3&2\\ 4&1&6&2\\ 2&2&3&6 \end{array} \rightarrow \begin{array}{cccc} 5&4&3&2\\ 2&2&3&6\\ 3&2&5&1\\ 4&1&6&2 \end{array} \rightarrow \begin{array}{cccc} 4&1&6&2\\ 2&2&3&6\\ 3&2&5&1\\ 5&4&3&2 \end{array} \rightarrow \begin{array}{cccc} 2&2&3&6\\ 3&2&5&1\\ 4&1&6&2\\ 5&4&3&2 \end{array}$$

l = št. števk

$T = O(l \cdot t_s)$ ... t_s = čas urejanja n števk

$U = \{0, 1, \dots, 9\}$, $|U| = M = 10$

$T = O(l \cdot t_s) = O(l \cdot (n+M))$ n elementov

$O(n+M)$ ... urejanje s štetjem

n-tiški sistem:

$5432 = 5 \cdot 10^3 + 4 \cdot 10^2 + 3 \cdot 10^1 + 2 \cdot 10^0$

$x \in [0, n^3-1]$

$x = a \cdot n^0 + b \cdot n^1 + c \cdot n^2$

$a, b, c \in \{0, 1, \dots, n-1\}$

$$\underbrace{\begin{array}{l} n^3 - 1 = [(n-1)][(n-1)][(n-1)] \\ x = (n-1)n^0 + (n-1)n^1 + (n-1)n^2 = n^3 - 1 \end{array}}_{l=3}$$

$$\underbrace{M = n}_{O(l \cdot (n+M))}$$

Pretvorba v n-tiški sistem: 3 mod operacije za en element!
Predpostavka: mod operacija vzame $O(1)$

Naj bo S mn. n poz. celih. št.
a) Naj alg., ki v času $O(n\log n)$ preveri, ali velja naslednji pogoj: $(\forall T\subseteq S)(\sum_{t\in T} t \geq |T|^3)$
Drugače: Če obstaja $T\subseteq S$, da velja $\sum_{t\in T} t < |T|^3$, alg. vrne false, sicer vrne true.
b) Poleg S in n je podan param. $1\leq k\leq n$. Napiši alg., ki v $O(n)$ času preveri, ali velja naslednji pogoj: $(\forall T\subseteq S)(|T| = k \Rightarrow \sum_{t\in T} t \geq |T|^3)$

a)
1) Uredimo S po velikosti:
$a_1, a_2, \dots, a_n$
Če $\underbrace{a_1 < 1^3}_{\text{se nikoli ne zgodi}}$, potem return false
nato $a_1 + a_2 < 2^3$ shrani si
nato $a_1 + a_2 + a_3 < 3^3$ -||-
⋮ ⋮ ⋮
$a_1 + \dots + a_k < k^3$
return true

Vaje 11

20.12.'24

Dana je $n \times m$ mat. A, ki sestoji iz 0 in 1. Poišči alg., ki vrne največji k, za katerega obstaja $k \times k$ podmat. samih 1 v mat. A.

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

$S[i,j]$ .... največji k, za katerega obstaja $k \times k$ podmat. samih 1 v mat. A in ima spodnji desni rob na polju (i,j).

$i,j \in \{1,\dots,n\}$

Končna rešitev: $\max_{i,j \in \{1,\dots,n\}} S[i,j]$

$i=1$: $S[1,j] = A[1,j] \quad \forall j$

$j=1$: $S[i,j] = A[i,1] \quad \forall i$

$$S = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \\ a_{n1} & \dots & & a_{nn} \end{bmatrix}$$

Poznamo 1 vrstico in 1 stolpec (isti kot v osnovni matriki)

$$S[i,j] = \begin{cases} 0 & \text{, če } A[i,j] = 0 \\ 1 + \min\{S[i,j-1], S[i-1,j], S[i-1,j-1]\} & \text{, če } A[i,j] \neq 1 \end{cases}$$

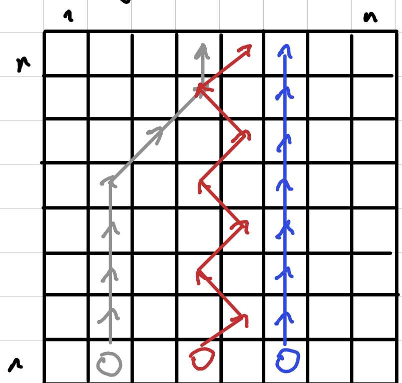
Izračun v konst. času ($O(n^2)$), prostorska zaht: $O(n^2)$

↳ lahko izboljšamo

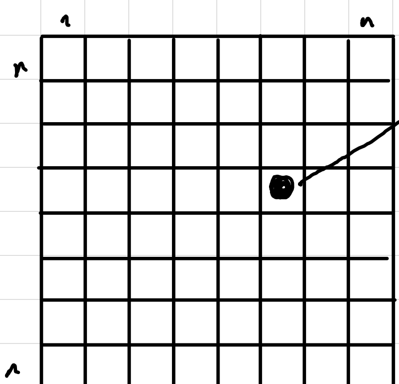
namesto hranjenja cele mat. S lahko hranimo le 2 vrstici in trenutni max. element

↳ če zač. mat. ne potrebujemo lahko prepisujemo elemente v začetni matriki

Dana je $n \times n$ šahovnica in kralj, ki je postavljen v spodnji vrstici. V $n-1$ potezah želi priti do zgornje vrstice, pri čemer premik s polja x na polje stane $c(x,y) > 0$. Napiši alg., ki poišče poteze kralja tako, da bo vsota cen vseh premikov najcenejša. (Kralja lahko postavimo kamorkoli na spodnjo vrstico in lahko konča kjerkoli na zg. vrstici)



Št. vseh dopustnih poter $\doteq 3^{n-1}$



$S[i,j]$ ... cena najcenejše poti kralja od sp. roba do polja (i,j). $i,j \in \{1,\dots,n\}$

Končna rešitev: $\min S[n,j],\ j \in \{1,\dots,n\}$

Trivialna rešitev v 1. vrstici: $S[1,j] = 0 \ \forall j$

$$\Rightarrow S[i,j] = \begin{cases} \min\{S[i-1,j-1]+c((i-1,j-1),(i,j)), \\ \quad S[i-1,j]+c((i-1,j),(i,j)), \\ \quad S[i-1,j+1]+c((i-1,j+1),(i,j))\}, & \text{če } j \neq n \wedge j \neq 1 \\ \min\{S[i-1,j]+c((i-1,j),(i,j)), \\ \quad S[i-1,j+1]+c((i-1,j+1),(i,j))\}, & \text{če } j = 1 \\ \min\{S[i-1,j-1]+c((i-1,j-1),(i,j)), \\ \quad S[i-1,j]+c((i-1,j),(i,j))\}, & \text{če } j = n \end{cases}$$