

Podatkovne strukture

So načini organizacije podatkov. (v programu):
seznam, tabele, slovarji, ...

Podatkovna struktura:

→ struktura podatkov

→ operacije na strukturi

Tabela $[x_0, x_1, \dots, x_n] = t$
(table, array) operacije:

- 1) Nova tabela velikosti n
- 2) vrni i -ti element
- 3) nastavi i -ti element na x
- 4) Vrni velikost tabele

$[None] * n$
 $[None \text{ for } i \text{ in range}(n)]$
 $t[i]$
 $t[i] = x$
 $len(t)$

Skład

(stack) $x_0, x_1, x_2, \dots, x_n$

Operacije:

- 1) Nov prazen sklad
- 2) Dodaj x na sklad → dodamo na konec
- 3) Vzemí s sklada → vzamemo s konca
- 4) Ali je prazen?

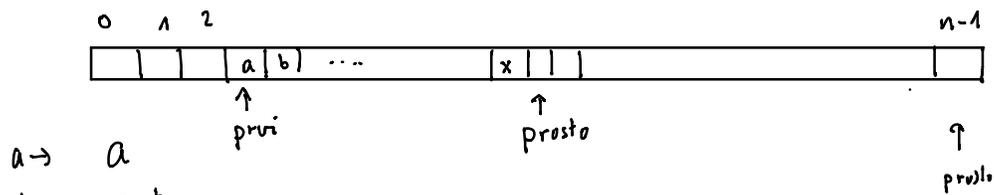
s. dodaj(x)
s. vzemi() → x

Vrsta: X_0, X_1, \dots, X_n

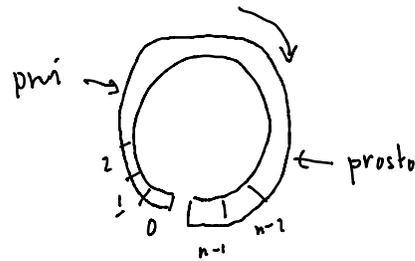
(queue)

- 1) Nova prazna vrsta
- 2) Dodaj x v vrsto (na konec)
- 3) Vzemij z vrste (z začetka)
- 4) Ali je vrsta prazna?

Kapaciteta = n



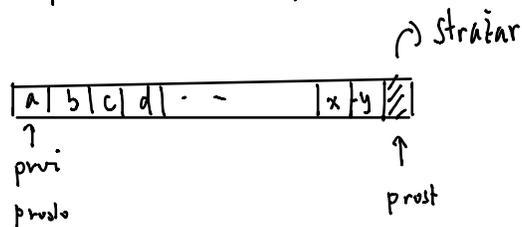
$a \rightarrow a$
 $b \rightarrow a b$
 $c \rightarrow a b c$
 $a \leftarrow b c$
 $d \rightarrow b c d$
 $b \leftarrow c d$



Objekt vrsta:

atributi: prvi, pravo, tabela

Polna?



↪ Stržinar

Kopica (prioritetna vrsta)

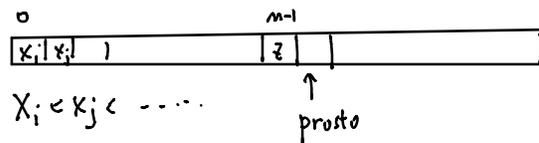
(heap, priority queue)

V kopici hranimo elemente x_0, x_1, \dots, x_n

- 1) Nova prazna kopica
- 2) Dodaj element na kopico
- 3) Vzemni najmanjši element s kopice

Naivna rešitev:

- podatke damo v tabelo, poskrbimo, da so vedno urejeni po velikosti



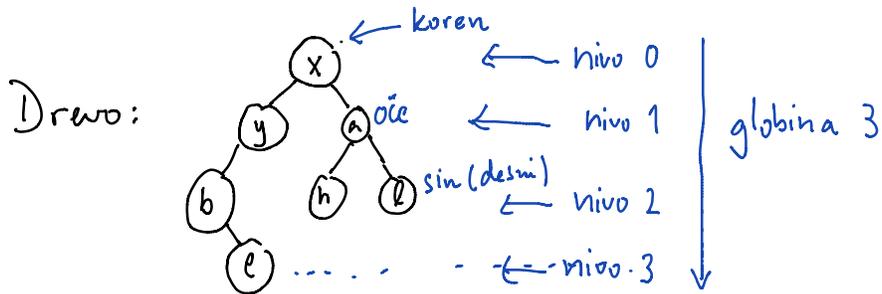
- dodajanje: dodamo element na konec in ga "potunkamo" (vstavimo) na pravo mesto $O(n)$

- vzemi prvega, premakni me ostale za 1 mesto na levo

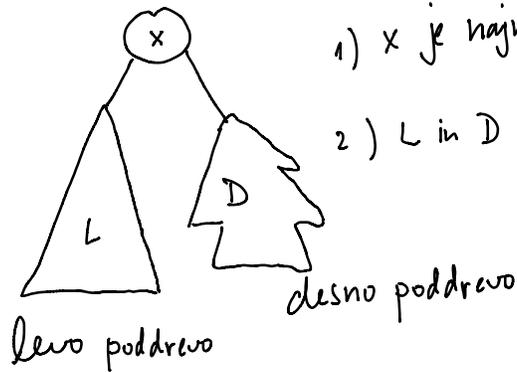
(izboljšava: prvi element ni nujno na indeksu 0) $O(1)$

$O(n)$

Kopica kot dvojiško drevo

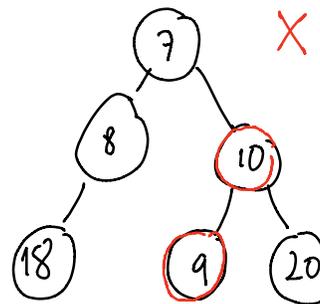
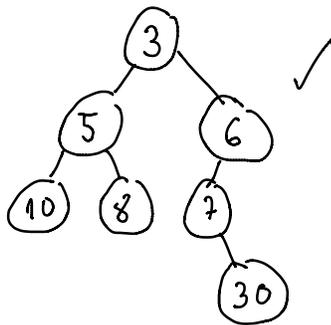


Kopica:

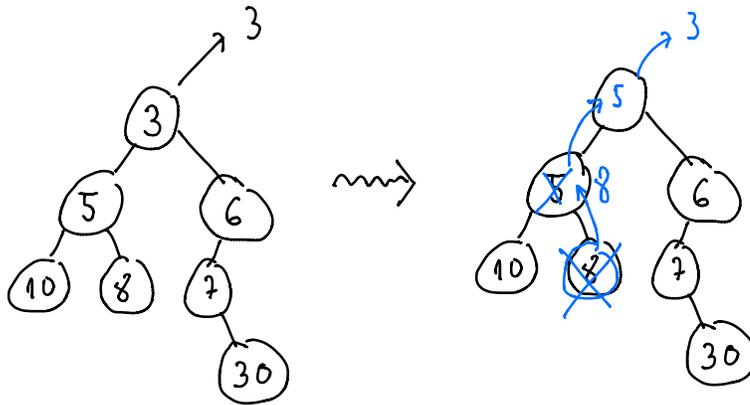


- 1) x je najmanjši element drevesa
- 2) L in D sta kopici

Primer:

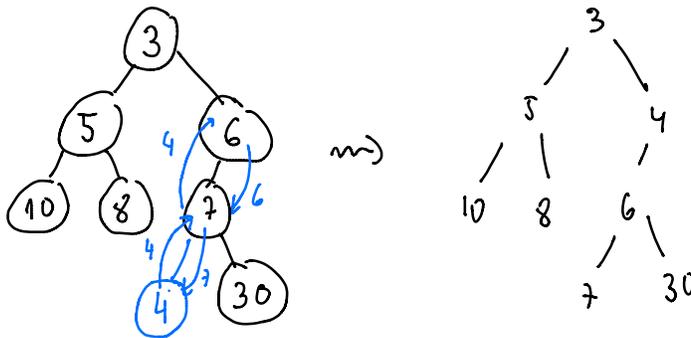


Kako vzamemo najmanjši element s kopice?



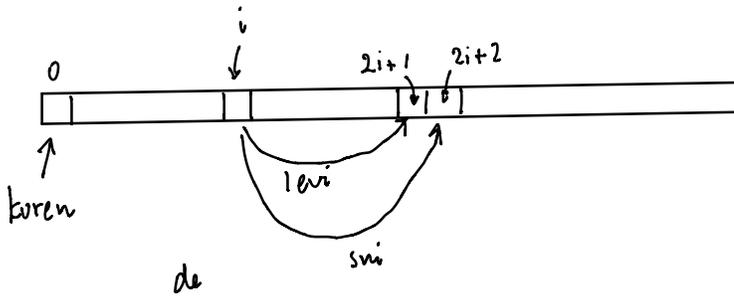
$O(\text{globina})$ korakov

Kako dodamo element?



$O(\text{globina})$

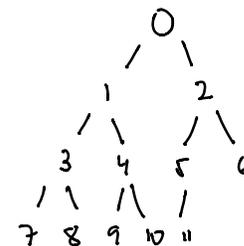
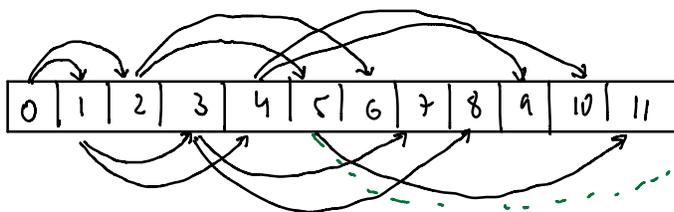
Drevo zložimo v tabelo:



Opomba: to deluje samo za

nekatera drevesa

↓
vsi nivoji, razen zadnjega,
so polni



Vedno dobimo drevo, ki je
skrajaj polno

Kopica hot tabela :

