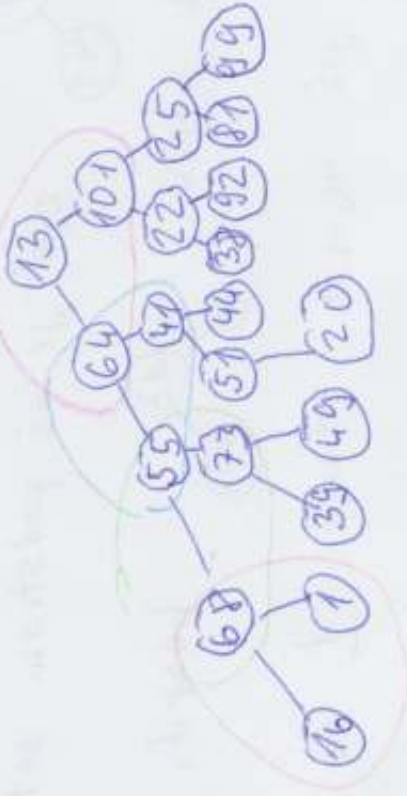


Petr Maronek 36456

Programování - Příklad 6.1. (b)

pole = 13, 64, 101, 55, 41, 22, 25, 68, 73, 51, 44, 38, 92, 81, 99, 16, 1, 39, 49, 20



- binární strom plnící věty zleva.

- Otec synové jsou v poli vždy ve vztahu $i \times 2 + 1$ - levý syn

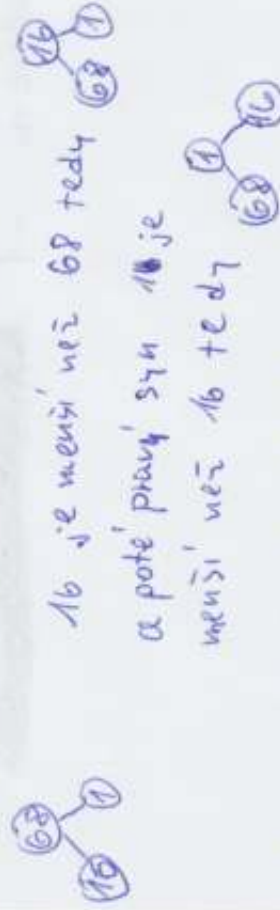
$i \times 2 + 2$ - pravý syn

• $7 \times 2 + 1 = 15$ (16)

• $7 \times 2 + 2 = 16$ (1)

sestupně třídění

heapify na nejvíce leví podstrom tedy



16 je menší než 68 tedy

a poté pravý syn 16 je

menší než 16 tedy

Pole je tedy přeskládané, když na

indexu 7 je 1, index 15 je 68

a index 16 je 16.

Tento podstrom je nyní v pořádku a pokračujeme dále tedy

výsledkem bude (55) (1) (73). Tento podstrom je setříděn a pokračujeme dále. =>



heapify va podstroim



opět vyměníme 1 s 64 tedy



da'll na nejvyšší podstrom, kde máme koreň.

```
graph TD; 13((13)) --- 7((7)); 13 --- 101((101))
```

heapify nám vrátí .


$$v_j' > |e_{de} k$$


Protože jsme již nejvíce vstoupili. 1 ze stromu odebereme.

v poli to znamená, že index pro průchod pole snížíme. Index pole 0, tedy 1, je setříděný. Strom však ztratí kořen, musíme tedy znovu zavolat funkci

neapify na vytvoření baldy nyní již 2 pole s indexem -1.