

1. Recapitulare analiza. Quicksort.

2. Heap-uri

3. Arboare binare de cautare

1. $P(z_i \text{ și } z_j \text{ să fie comparate}) = \frac{2}{j-i+1}$

exemplu

5 9 23 14 24 8 7

$$z_1 = 5$$

$$z_2 = 7$$

$$z_3 = 8$$

$$z_4 = 9$$

$$z_5 = 14$$

$$z_6 = 23$$

$$z_7 = 24$$

→ pivot 14: 5 9 8 7 14 23 24

5 7 8 9

2. Heap-uri (MAX-heap / MIN-heap)

- afisare min / max $O(1)$

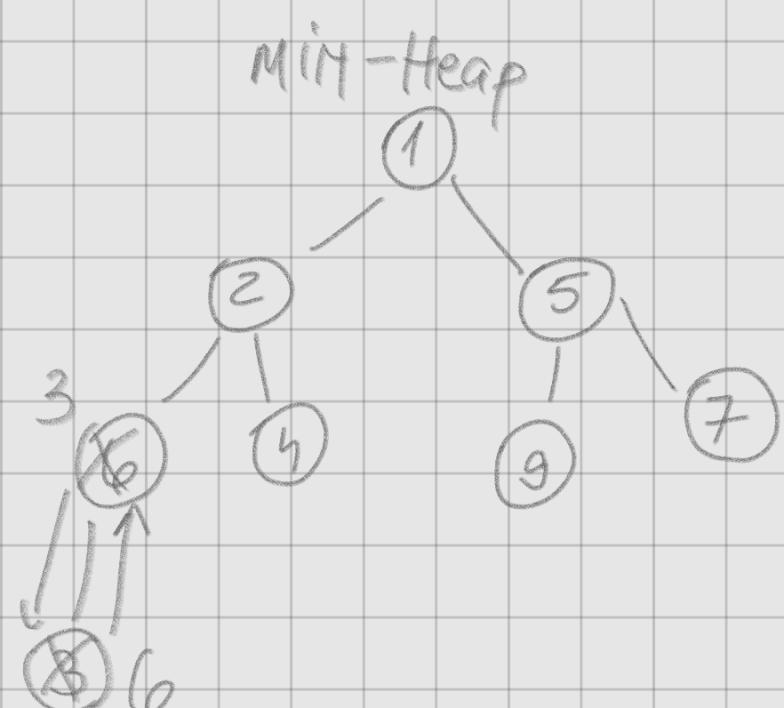
- extragere min / max $O(\log n)$

- inserare element $O(\log n)$

no heart left to break
no tears left to cry.

and I'm only 20.

- Heap-ul să fie în visualizare ca pe un arbore binar complet, dar îl reprez. Ca vector
- Proprietatea principală este valoarea din fiecare nod este mai mică (Min-Heap) / mare (Max-Heap) decât toate elementele din subarbore



Heap-Sort

- n inserări în Heap $\rightarrow O(n \log n)$
- extrageri min/max $\rightarrow O(n \log n)$

Pb: se dă k liste sortate ce au în total n nr. Să se sorteze cele n numere

ex: $k=3$ $n=10$

2 4 8

3 6 12 15 24

7 10

2 3 4 6 7 8 10 12 14 24

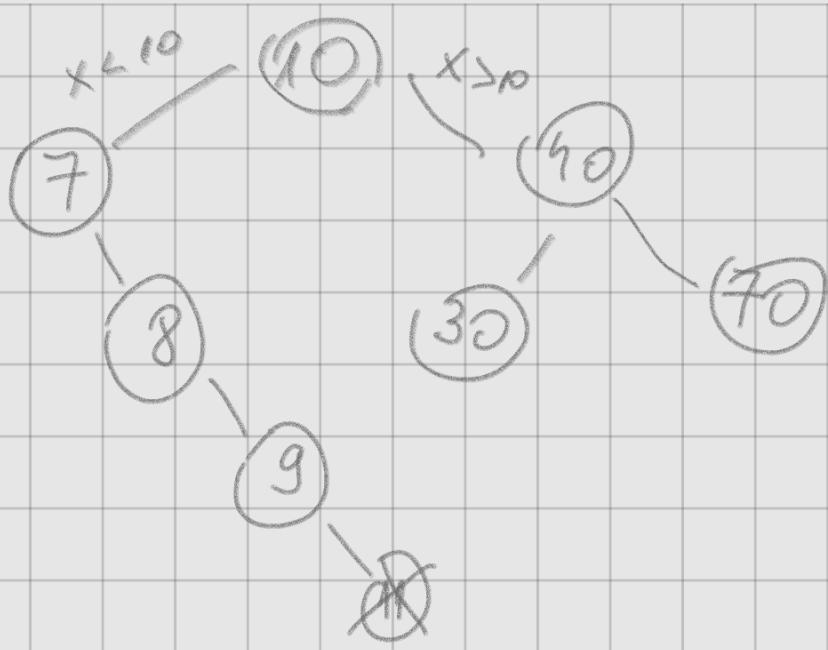
0. quicksort ($n \log n$)

1. k interclasări $O(kn)$



Arbore binar de căutare

Def Un arbore binar de căutare este un arbore binar în care valoarea asociată fiecărui nod este mai mare decât toate nodurile din subarborele stâng și mai mică decât toate nodurile din subarborele drept.



11 nu e mai mic decât 10

- căutare $O(n)$
- inserție $O(n)$
- min $O(n)$
- max $O(n)$
- succesor $O(n)$ \rightarrow cel mai mic nr mai mare decât x
 \hookrightarrow x are fiu drept
- predecesor $O(n)$ \hookrightarrow următor pe arbore până dăm de fiu stâng
- ștergere $O(n)$