

1. Heap-uri (Heap Sort)

2. Arbori binari de căutare

Heap-uri

a) Ce operații suportă?

a.1) Inserția unui element $O(\log n)$

a.2) extragerea minimului / maximumului $O(\log n)$

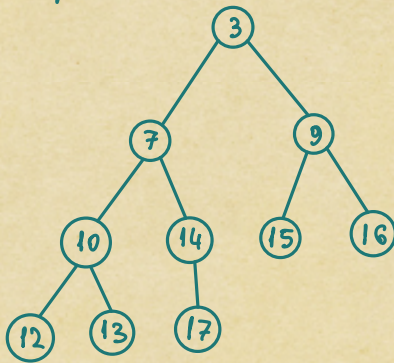
a.3) returnarea minimului / maximumului $O(1)$

Def: Un heap este un arbore binar plin (fiecare nod are 2 fii, cu excepția, posibil, a penultimului nivel) cu proprietatea că valoarea din fiecare nod este mai mare (mai mică) decât toate valorile din subarbore.

Min-heap \Rightarrow fiecare nod va fi mai mic decât toate nodurile din subarbore.

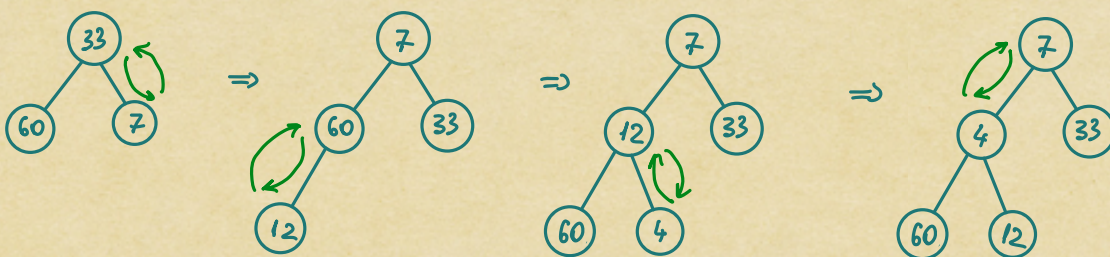
Max-heap \Rightarrow fiecare nod va fi mai mare decât toate nodurile din subarbore.

Exemplu: (min-heap)



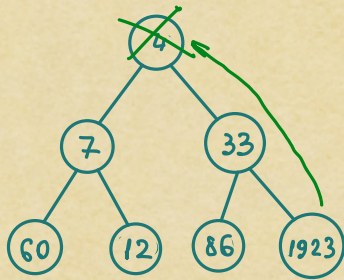
⊙ proprietate a heap-ului este că rădăcina va stoca minimul (min-heap) sau maximumul (max-heap).

Exemplu: 33, 60, 7, 12, 4, 86, 1923

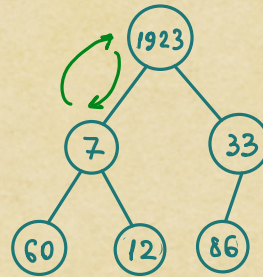




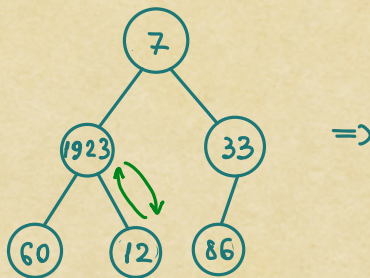
Exemplu:



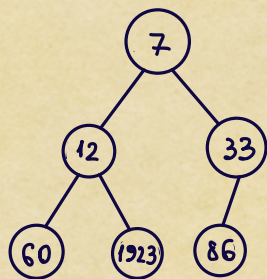
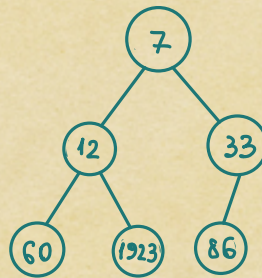
⇒



⇒



⇒



Sub formă de vector

	1	2	3	4	5	6	7
V	7	12	33	60	1923	86	x

i

- fiu stâng $2*i$
- fiu drept $2*i+1$
- tată $[i/2]$

Exemplu: 33, 60, 7, 12, 4, 86

1	
33	

1	2	3	
33	60	7	

⇒

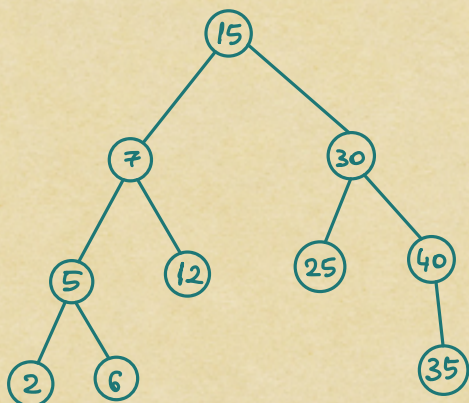
7	60	33	



Arbori binari de căutare

Def: Un arbore binar de căutare este un arbore binar cu proprietatea că valoarea asociată fiecărui nod este mai mare decât toate valorile din subarborul stâng și mai mică decât toate valorile din subarborul drept.

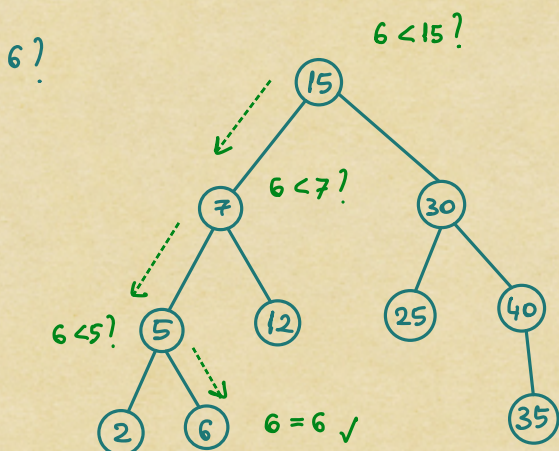
Exemplu:



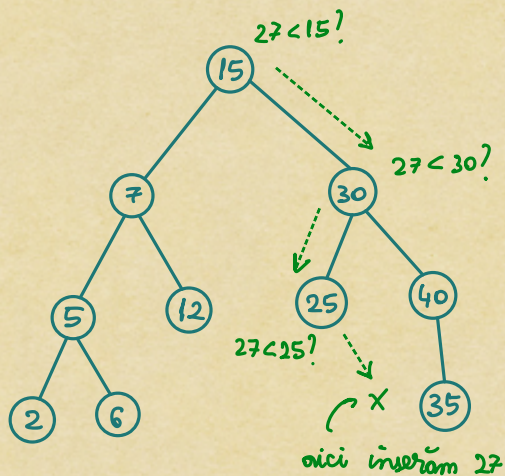
Operații:

1. Inserție $O(n)$
2. Căutare $O(n)$
3. Min/Max $O(n)$
4. Succesor / Predecesor $O(n)$
5. Ștergere

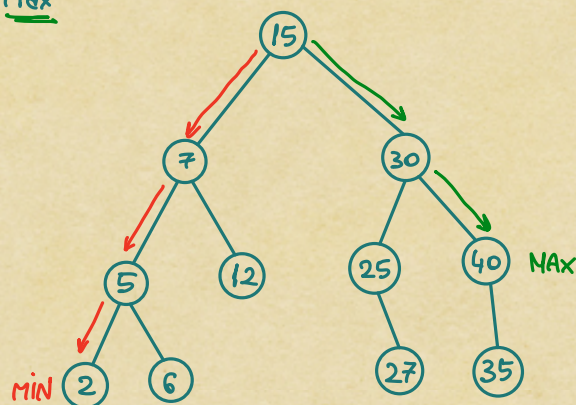
Ex: Căutare



27?

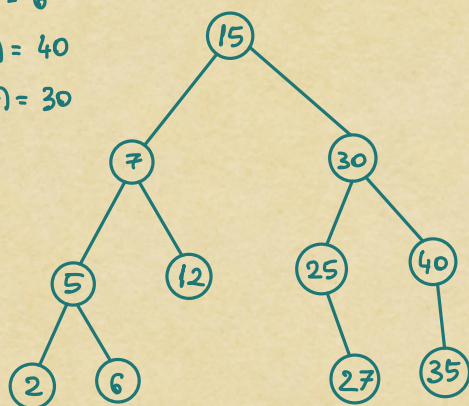


Ex Min / Max



Successorul unui nod este nodul cu valoarea cea mai mică, dar, în același timp, mai mare decât x .

Ex Successor(5) = 6
Successor(35) = 40
Successor(27) = 30



Predecesorul —||— cea mai mare —||— mai mică decât x .

Ex Predecesor(7) = 6 Predecesor(6) = 5 Predecesor(27) = 25

Successor x: a) Min din subarborele drept dacă x are fiu drept

b) Dacă x nu are fiu drept, cel mai mic strămoș al cărui fiu stâng este tot strămoș al lui x

Ștergerea

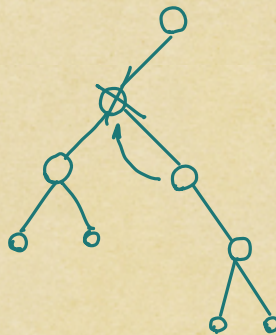
a) frunză



b) Nodul nu are fiu drept



c) Vrem să ștergem un nod z care are fiu drept y, dar y nu are fiu stâng



d) Nodul z are fiu drept y care are fiu stâng

