

HeapSort

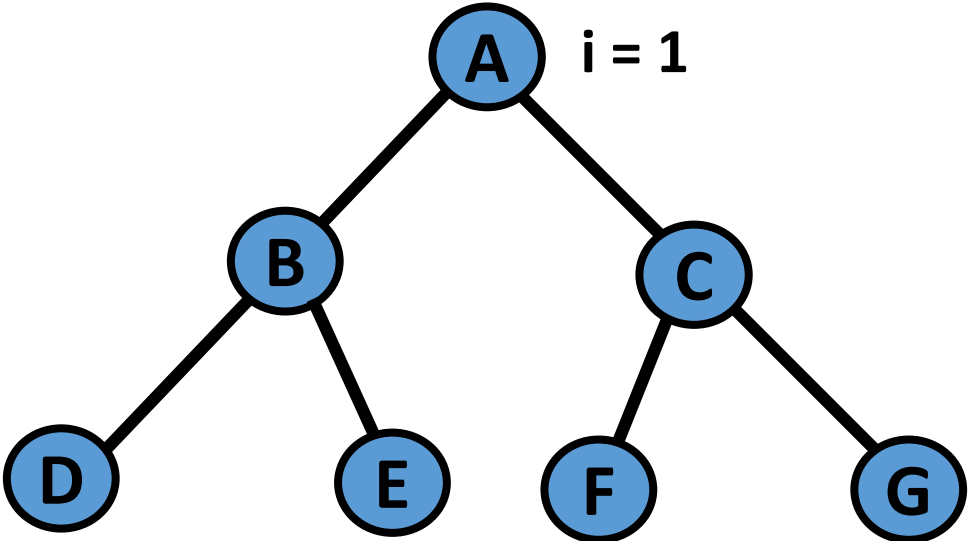
Cuprins

- Reprezentarea arborilor binari folosind array-ul;
- Heap Max & Min;
- Inserarea elementelor în Heap (Max);
- Ștergerea elementelor din Heap (Max);
- Heapsort;
- Heapify;
- Priority Queues.

Arbori binari

Arbore binar complet

Nu are elemente necomplete în tablou

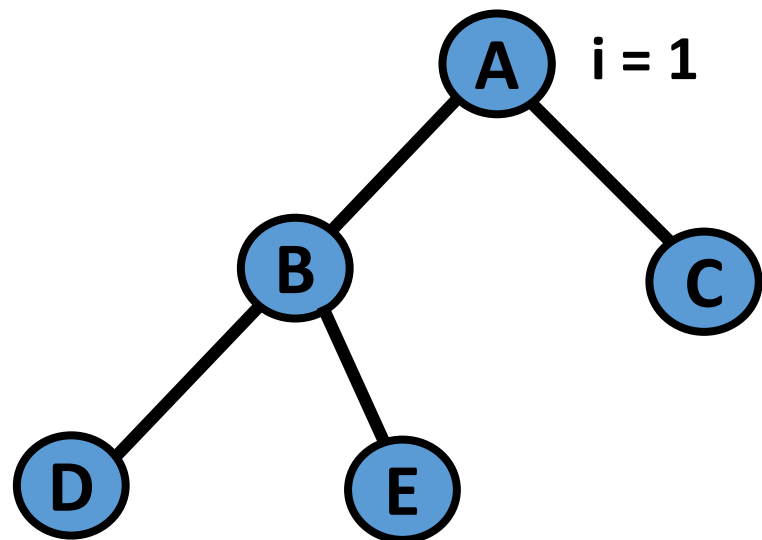


1	2	3	4	5	6	7
A	B	C	D	E	F	G

Nodul	indexul
Un nod	i
Copilul din stânga a nodului	$2*i$
Copilul din dreapta a nodului	$2*i+1$
Părintele nodului	$i/2$

A	$i = 1$
B	$i = 2, \quad 2 * i(A) = 2 * 1 = 2$
C	$i = 3, \quad 2 * i(A) + 1 = 2 * 1 + 1 = 3$
D	$i = 4, \quad 2 * i(B) = 2 * 2 = 4$
E	$i = 5, \quad 2 * i(B) + 1 = 2 * 2 + 1 = 5$
F	$i = 6, \quad 2 * i(C) = 2 * 3 = 6$
G	$i = 7, \quad 2 * i(C) + 1 = 2 * 3 + 1 = 7$

Arbori binari



Nodul	indexul
Un nod	i
Copilul din stânga a nodului	$2*i$
Copilul din dreapta a nodului	$2*i+1$
Părintele nodului	$i/2$

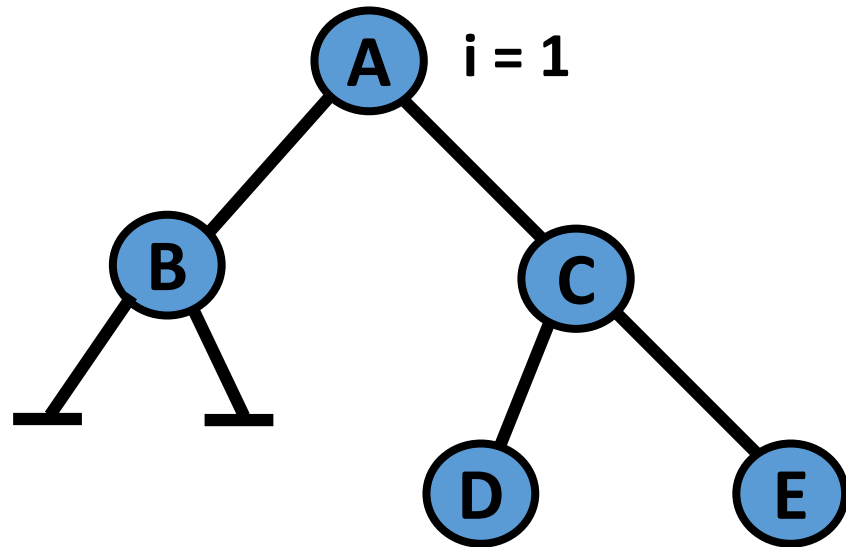
Arbore binar complet

Nu are elemente necomplete în tablou

1	2	3	4	5
A	B	C	D	E

A	$i = 1$
B	$i = 2, \quad 2 * i(A) = 2 * 1 = 2$
C	$i = 3, \quad 2 * i(A) + 1 = 2 * 1 + 1 = 3$
D	$i = 4, \quad 2 * i(B) = 2 * 2 = 4$
E	$i = 5, \quad 2 * i(B) + 1 = 2 * 2 + 1 = 5$

Arbori binari



Nodul	indexul
Un nod	i
Copilul din stânga a nodului	$2*i$
Copilul din dreapta a nodului	$2*i+1$
Părintele nodului	$i/2$

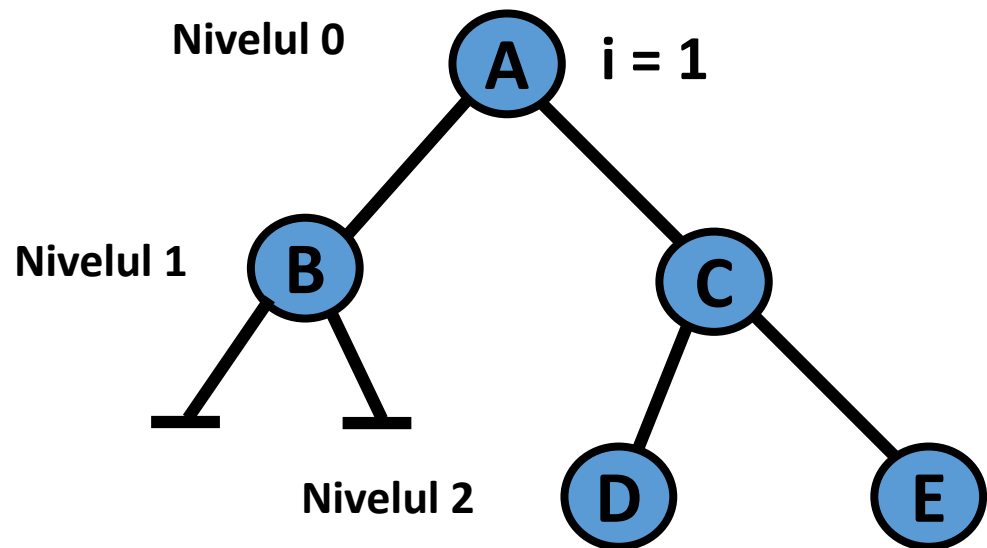
Arbore binar necomplet

Are elemente necompletate în tablou, index 4 și 5

1	2	3	4	5	6	7
A	B	C	-	-	D	E

A	$i = 1$
B	$i = 2, \quad 2 * i(A) = 2 * 1 = 2$
C	$i = 3, \quad 2 * i(A) + 1 = 2 * 1 + 1 = 3$
D	$i = 6, \quad 2 * i(C) = 2 * 3 = 6$
E	$i = 7, \quad 2 * i(C) + 1 = 2 * 3 + 1 = 7$

Arbori binari



Nodul	indexul
Un nod	i
Copilul din stânga a nodului	$2*i$
Copilul din dreapta a nodului	$2*i+1$
Părintele nodului	$i/2$

Arbore binar necomplet

Are elemente necompletate în tablou, index 4 și 5

1	2	3	4	5	6	7
A	B	C	-	-	D	E

A	$i = 1$
B	$i = 2, \quad 2 * i(A) = 2 * 1 = 2$
C	$i = 3, \quad 2 * i(A) + 1 = 2 * 1 + 1 = 3$
D	$i = 6, \quad 2 * i(C) = 2 * 3 = 6$
E	$i = 7, \quad 2 * i(C) + 1 = 2 * 3 + 1 = 7$

IMPORTANT!

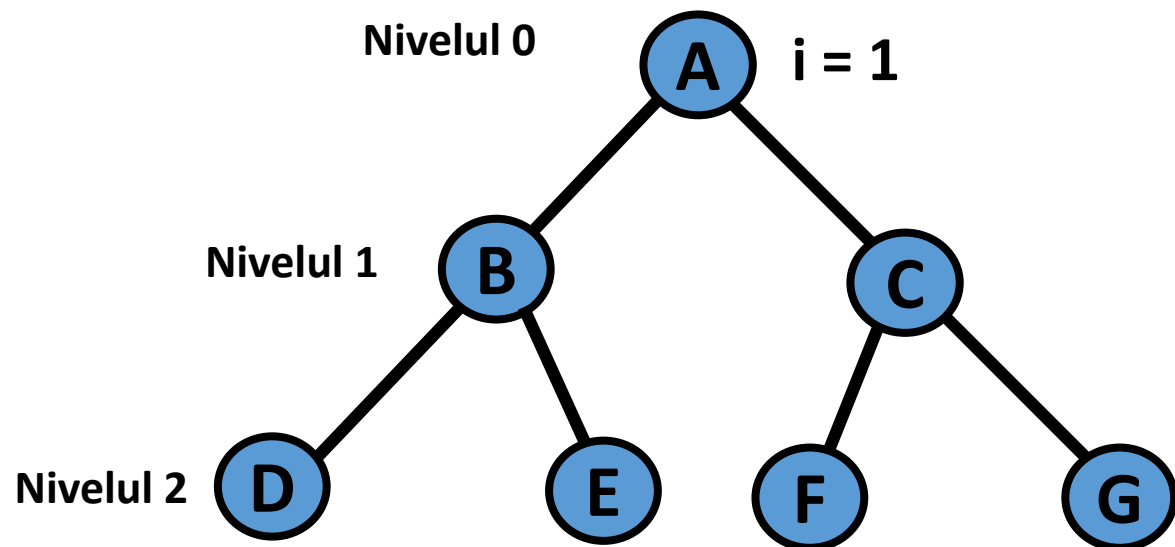
Un arbore poate avea maximum $2^{h+1} - 1$ noduri, unde h este înălțimea* arborelui, $2^{2+1} - 1 = 7$.

*Înălțimea arborelui – numărul de nivel maxim asociat nodurilor terminale.

Arbori binari

Arbore binar complet

Nu are elemente necompletate în tablou

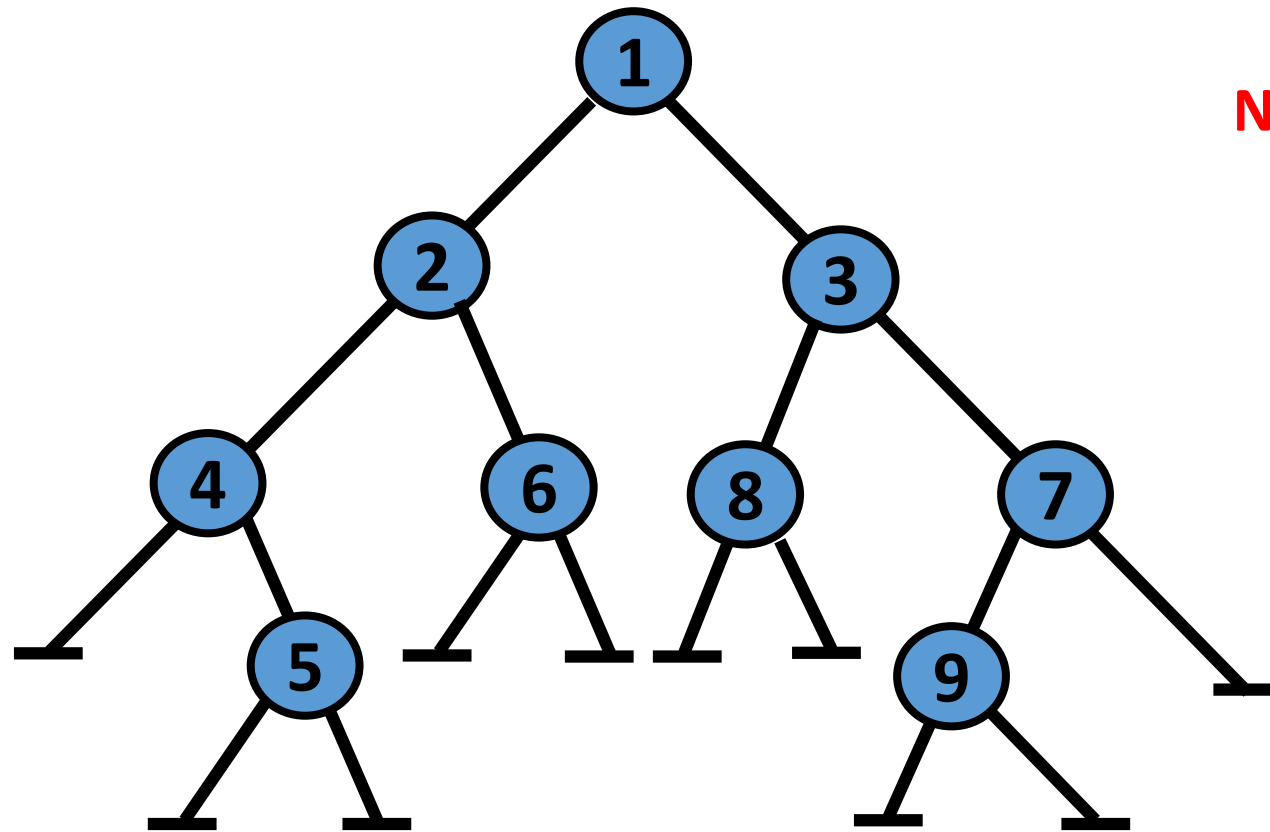


1	2	3	4	5	6	7
A	B	C	D	E	F	G

IMPORTANT!

Un arbore își completează elementele pe nivele(linie). Pentru a avea un arbore complet, verificăm nodurile de la stânga spre dreapta pe linie. Numărul de elemente pe nivel este de 2^n , unde n este nivelul pe care se află.

Arbori binari: Ce fel de arbore avem, complet sau nu?

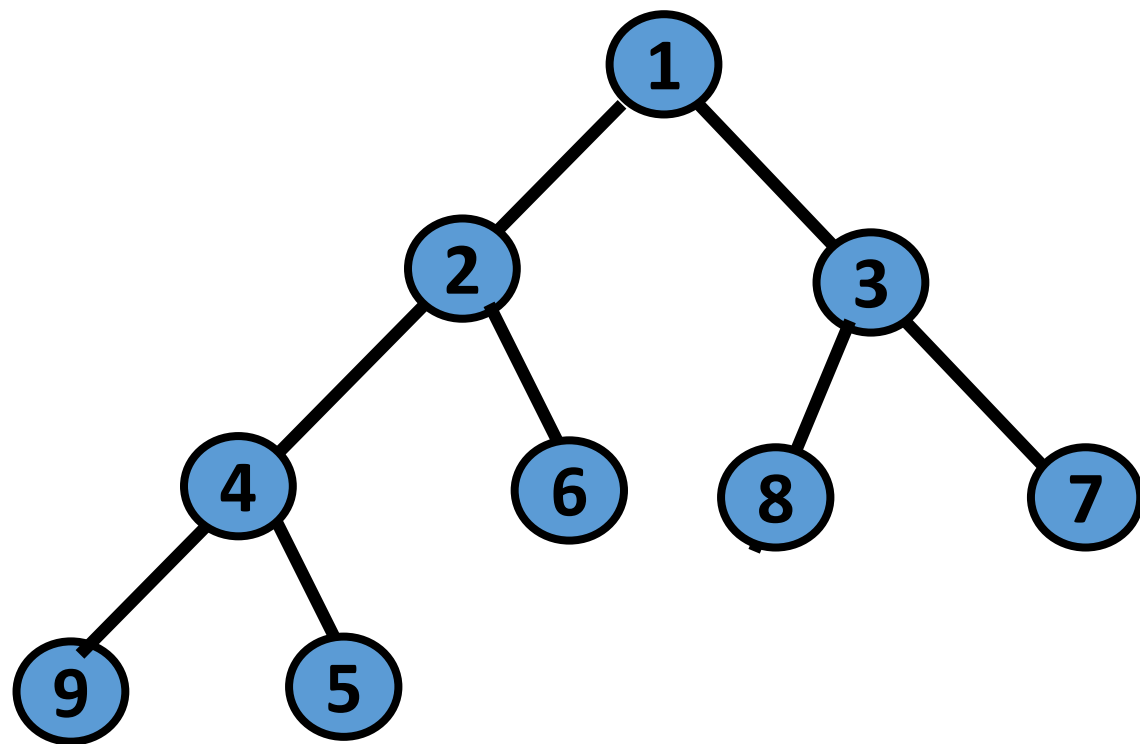


Necomplet

IMPORTANT!

Un arbore își completează elementele pe nivele(linie). Pentru a avea un arbore complet, verificăm nodurile de la stânga spre dreapta pe linie. Numărul de elemente pe nivel este de 2^n , unde n este nivelul pe care se află.

Arbori binari: Ce fel de arbore avem, complet sau nu?

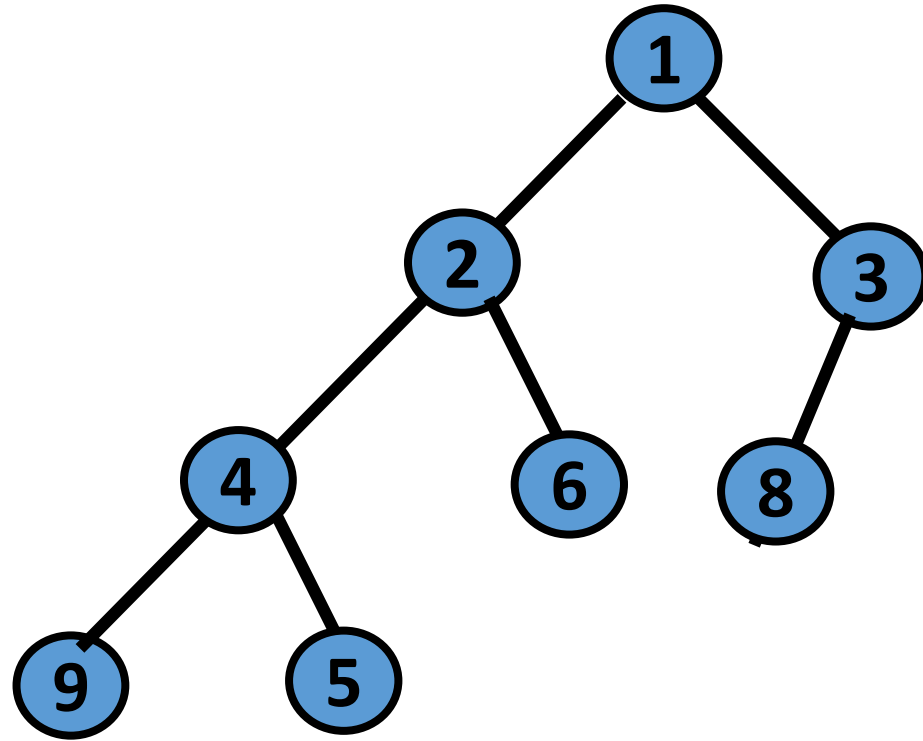


Complet

IMPORTANT!

Un arbore își completează elementele pe nivele(linie). Pentru a avea un arbore complet, verificăm nodurile de la stânga spre dreapta pe linie. Numărul de elemente pe nivel este de 2^n , unde n este nivelul pe care se află.

Arbori binari: Ce fel de arbore avem, complet sau nu?



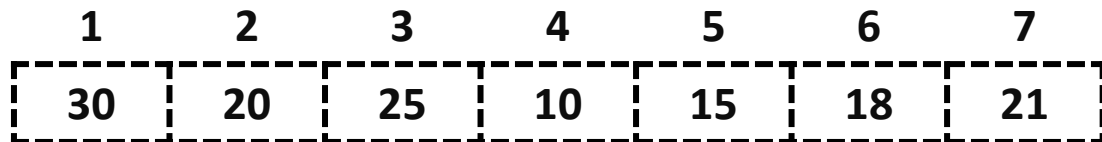
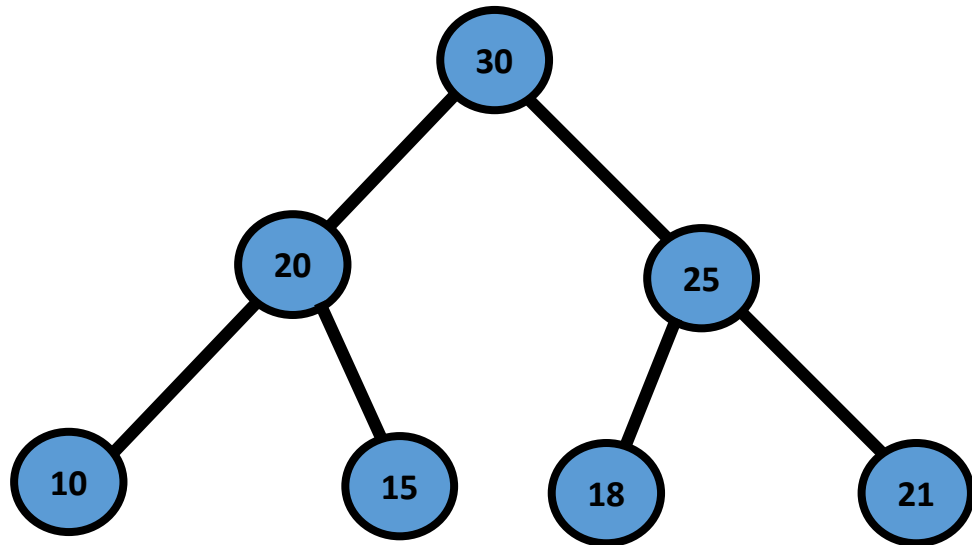
Necomplet

IMPORTANT!

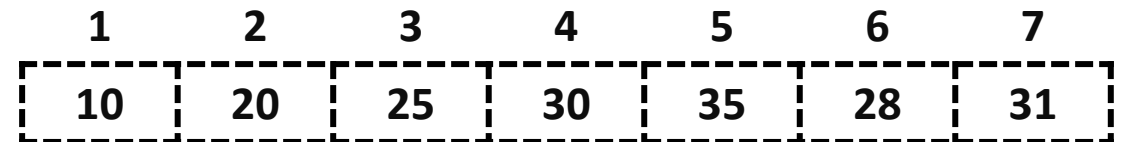
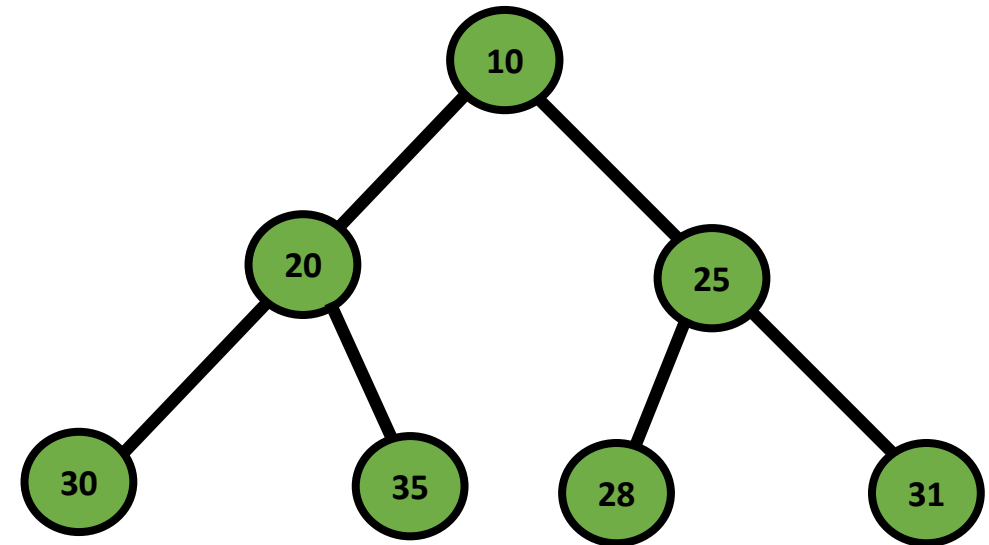
Un arbore își completează elementele pe nivele(linie). Pentru a avea un arbore complet, verificăm nodurile de la stânga spre dreapta pe linie. Numărul de elemente pe nivel este de 2^n , unde n este nivelul pe care se află.

Heap Max & Min

Max Heap



Min Heap

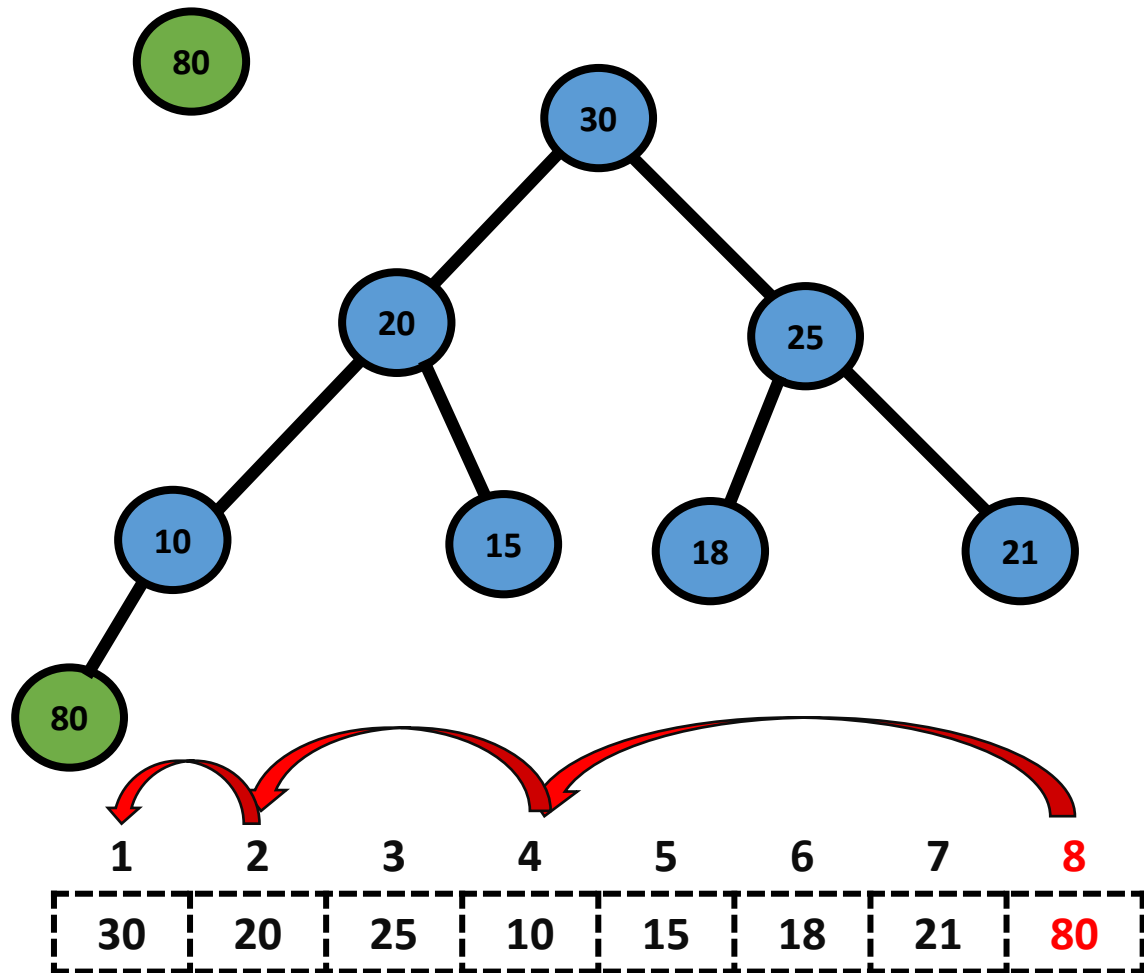


IMPORTANT!

Un MaxHeap/MinHeap este format din noduri a căror valoare este mai mare/mică decât valoarea descendenților(copiilor) acestuia.

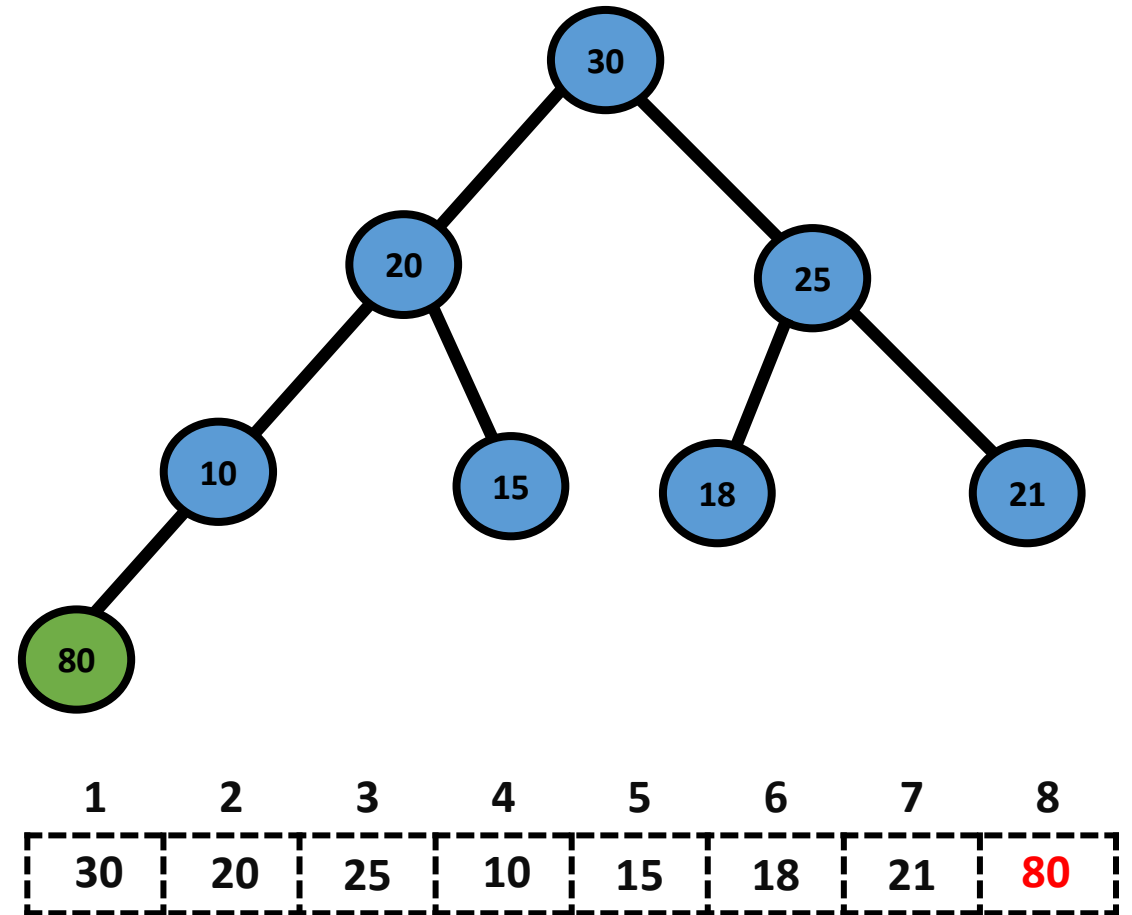
Inserarea elementelor în Heap (Max)

Max Heap



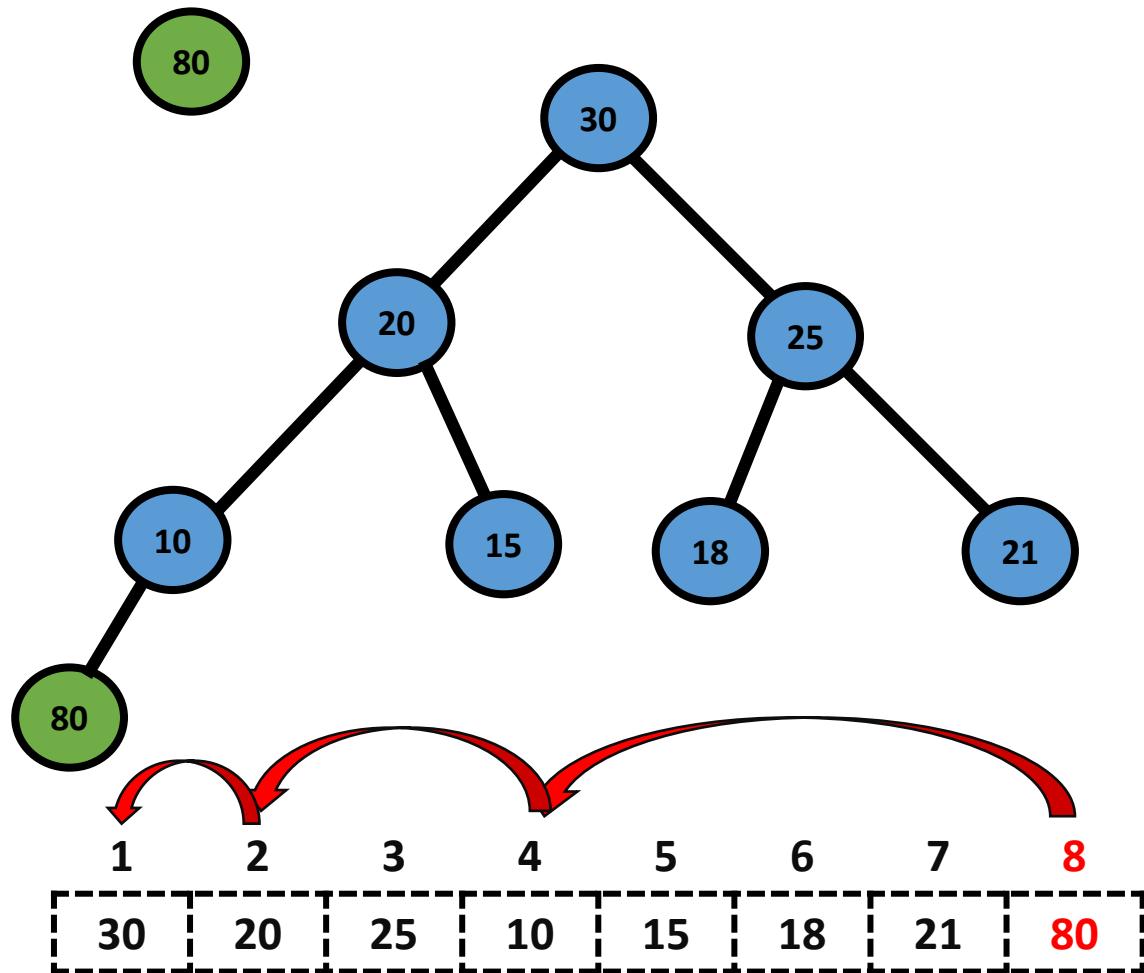
Părintele lui 80: $i/2 = 8/2 = 4$

Max Heap



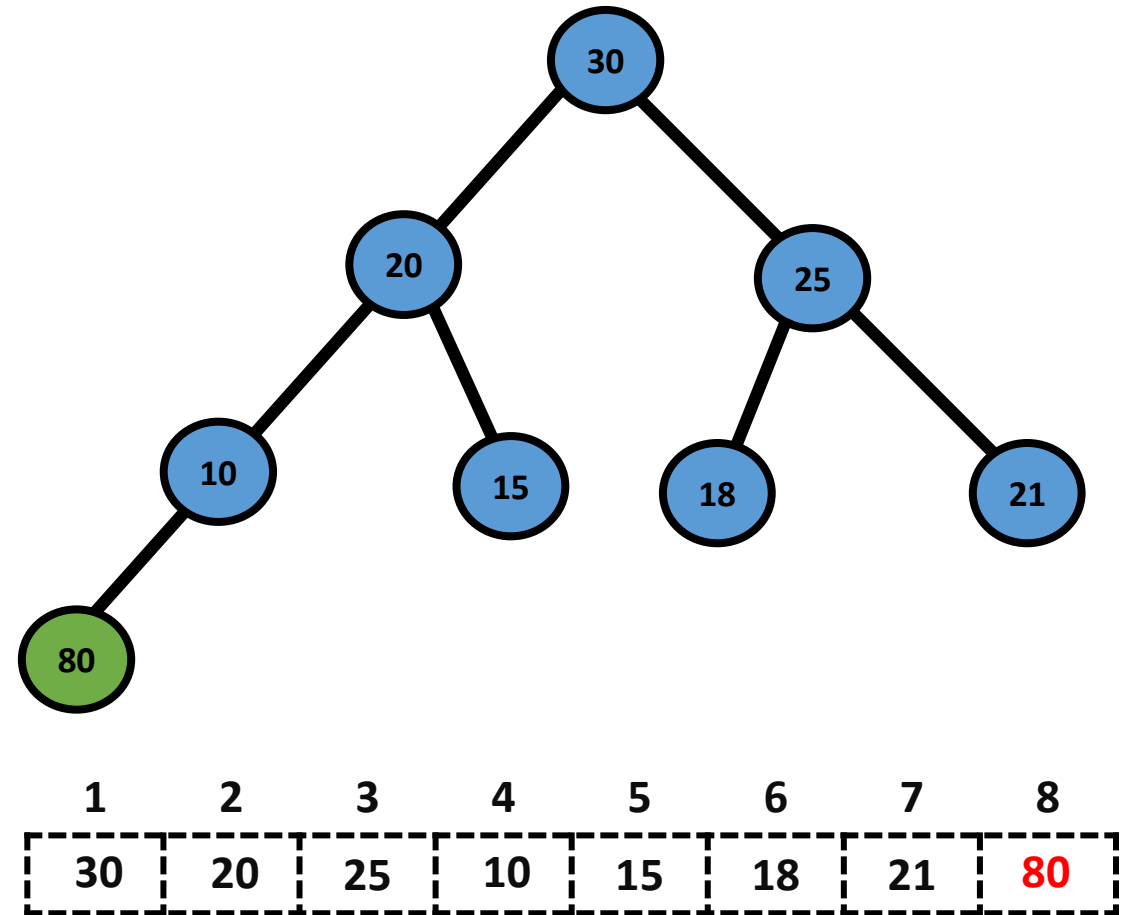
Inserarea elementelor în Heap (Max)

Max Heap

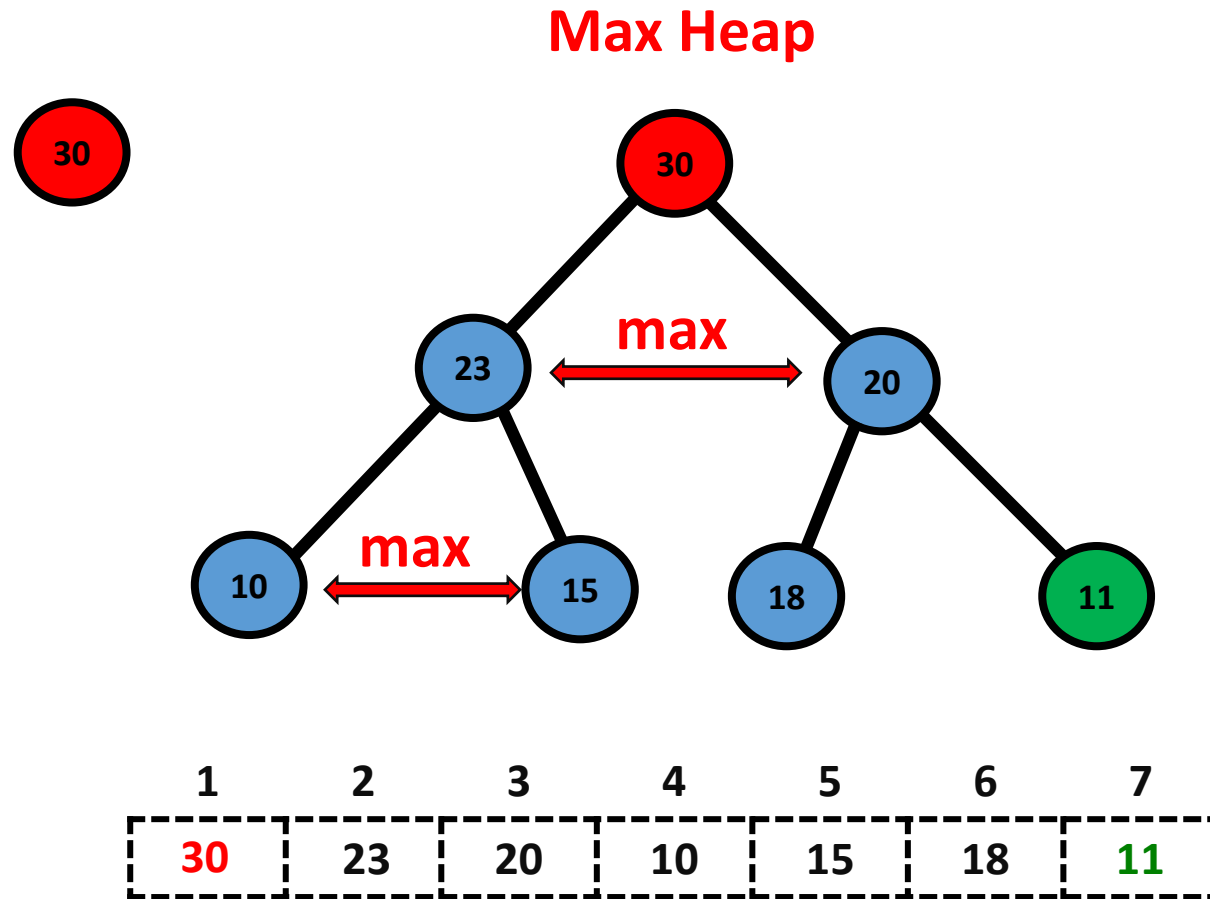


Părintele lui 80: $i/2 = 8/2 = 4$

Max Heap



Ștergerea elementelor din Heap (Max)

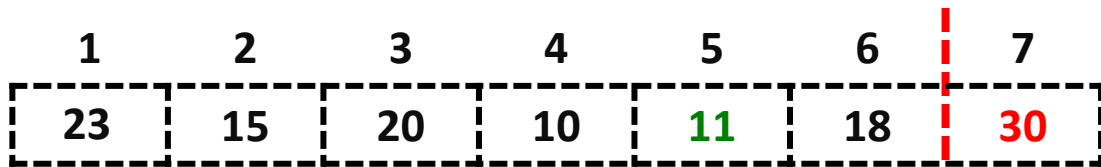
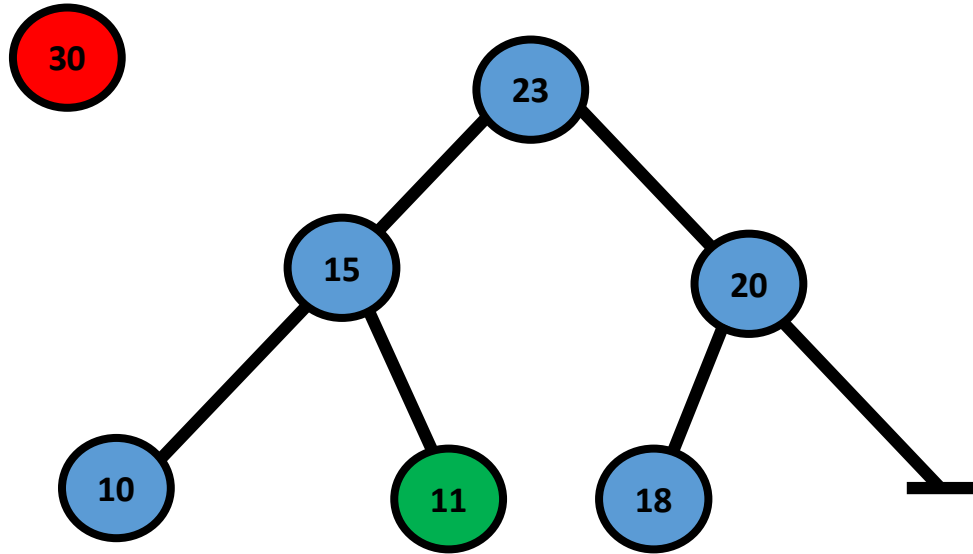


IMPORTANT!

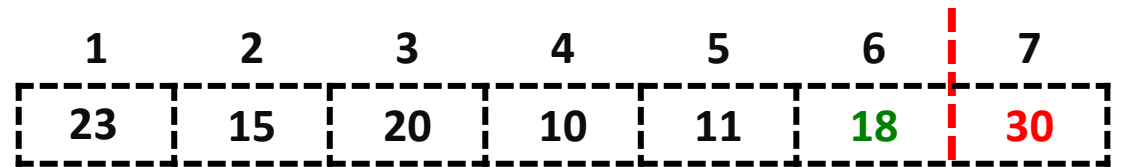
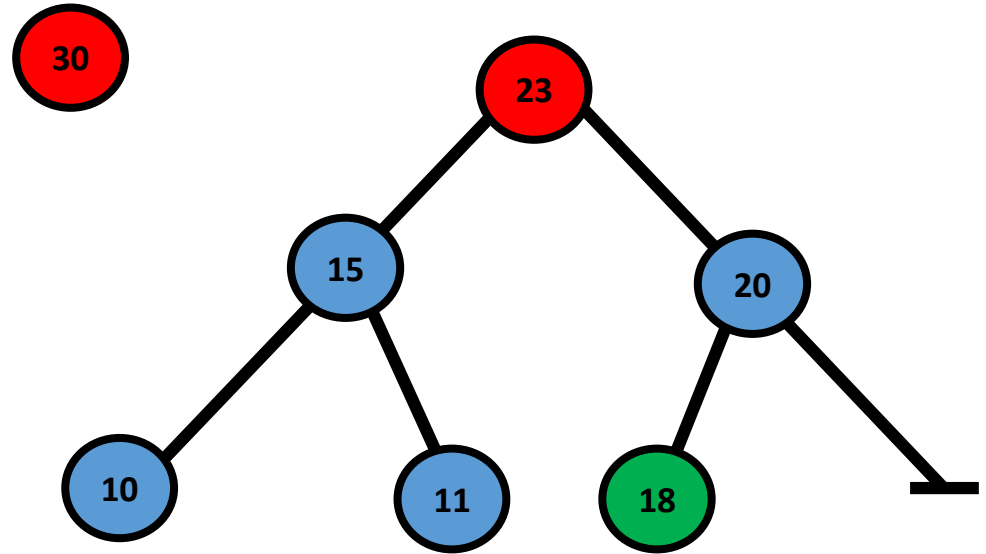
Ștergerea unui element din heap se va face din vârful piramidei, adică se va șterge rădăcina arborelui binar.

Ștergerea elementelor din Heap (Max)

Max Heap

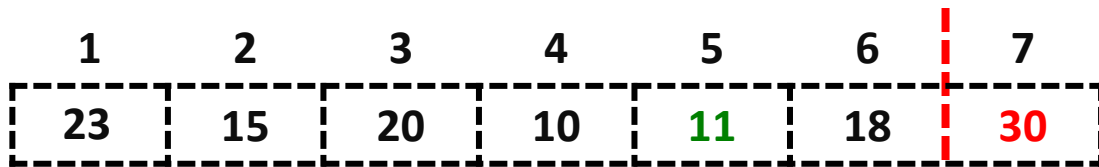
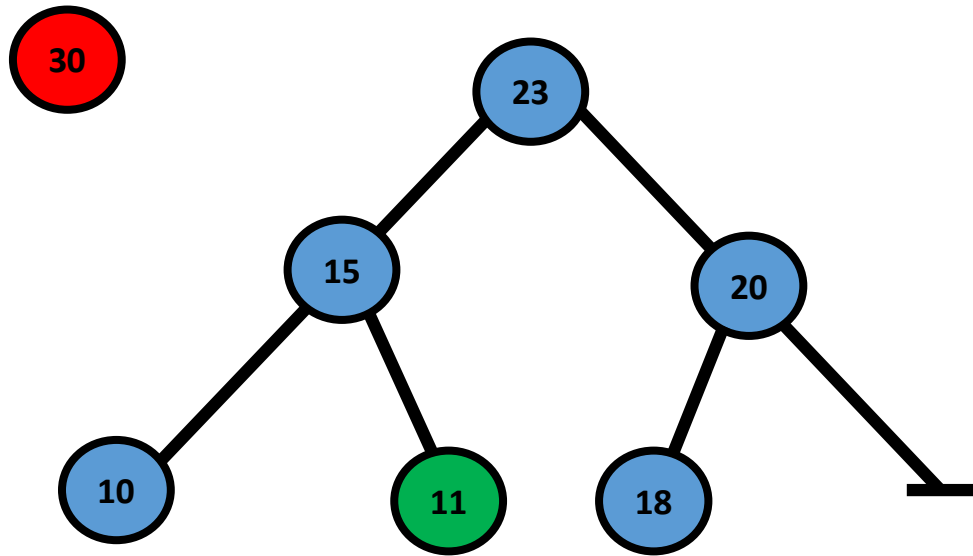


Max Heap

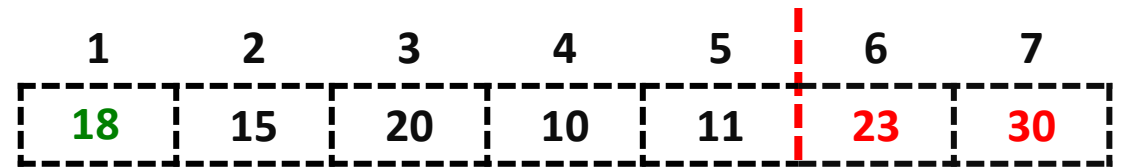
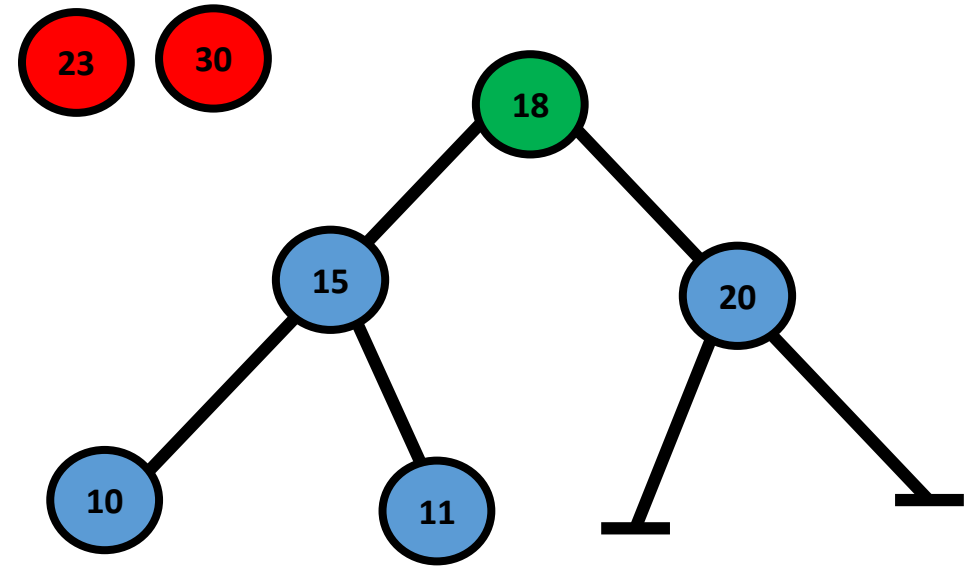


Ștergerea elementelor din Heap (Max)

Max Heap

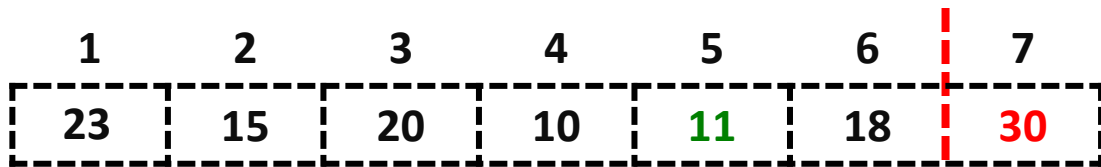
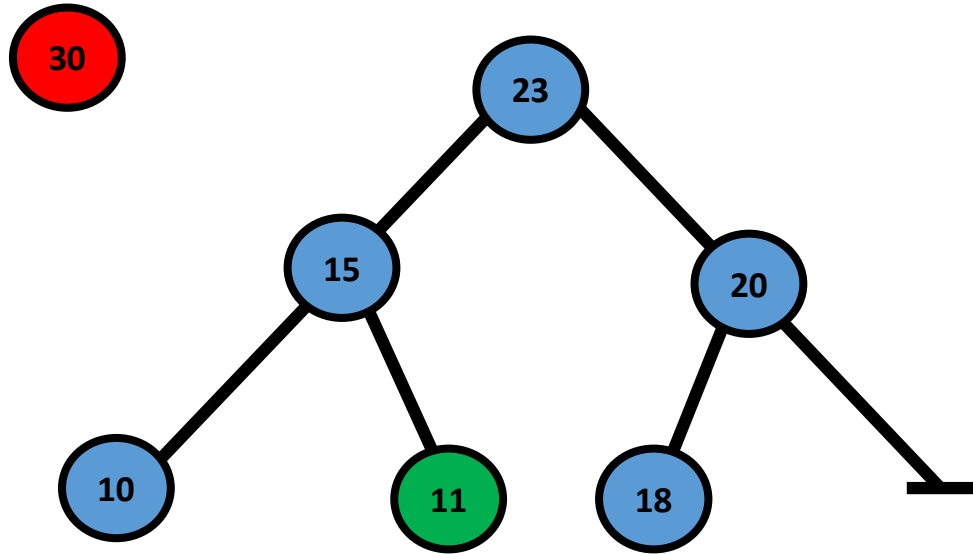


Max Heap

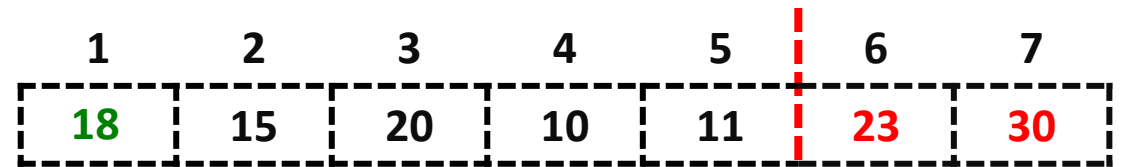
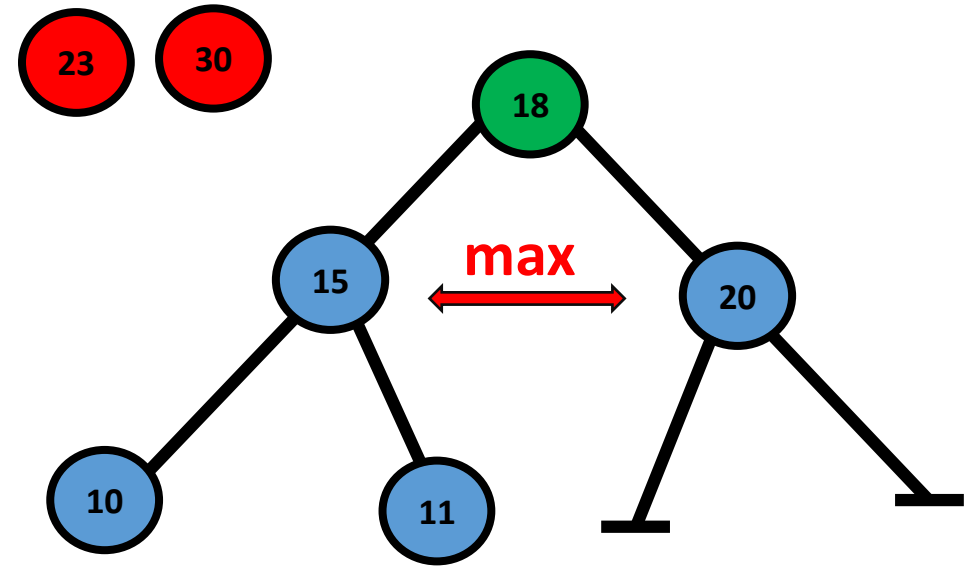


Ștergerea elementelor din Heap (Max)

Max Heap

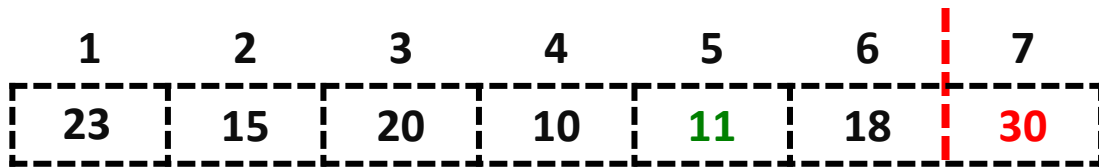
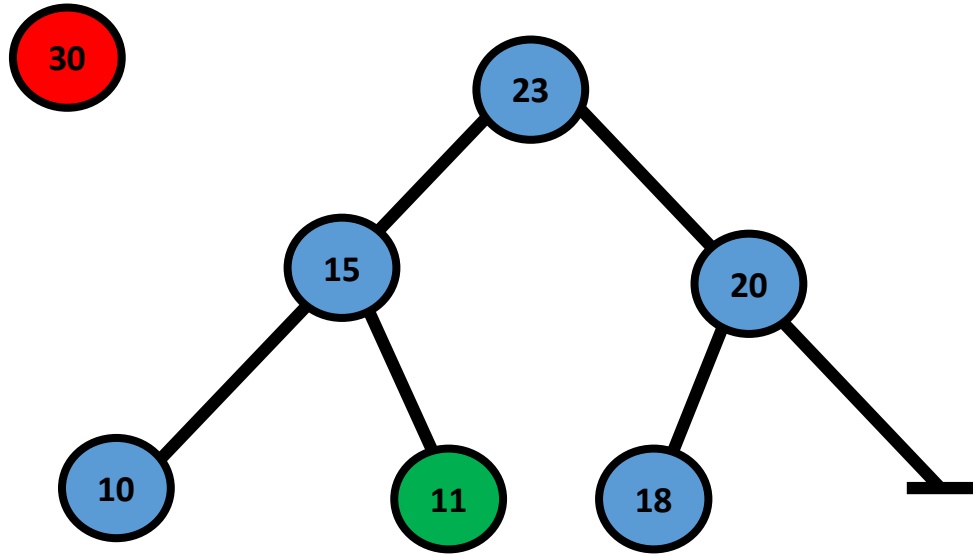


Max Heap

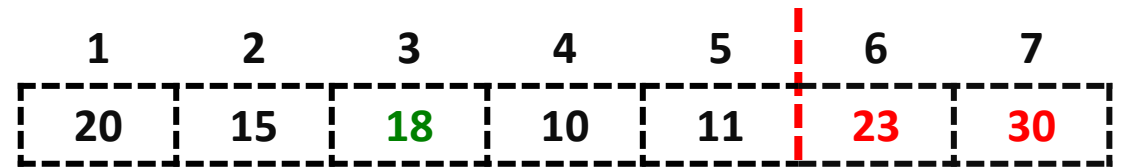
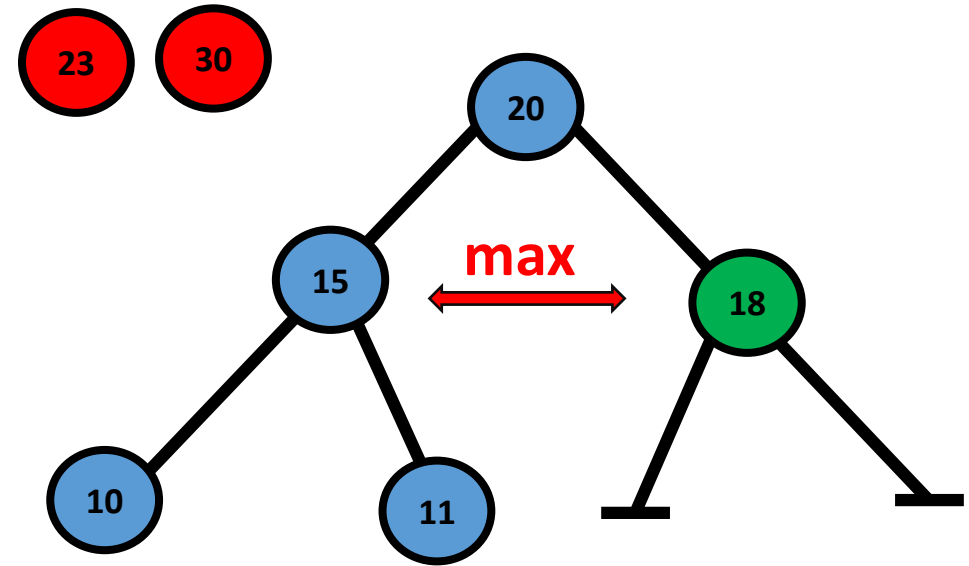


Ștergerea elementelor din Heap (Max)

Max Heap

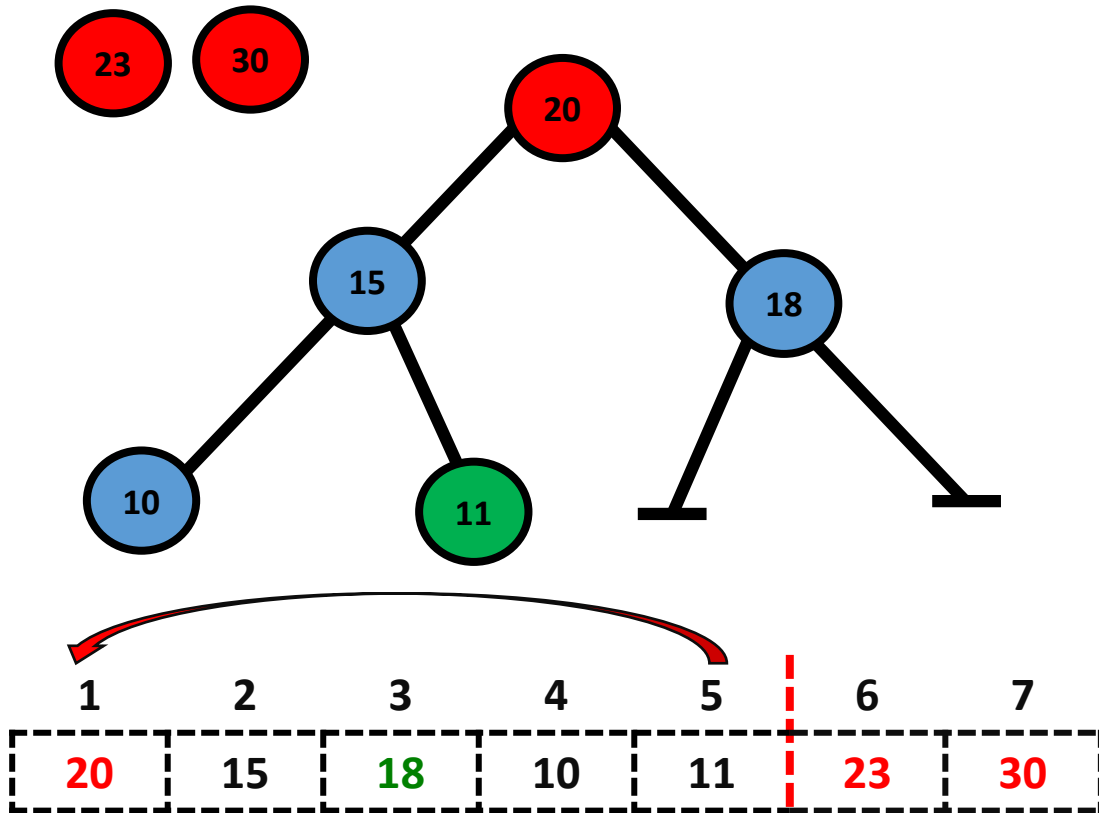


Max Heap

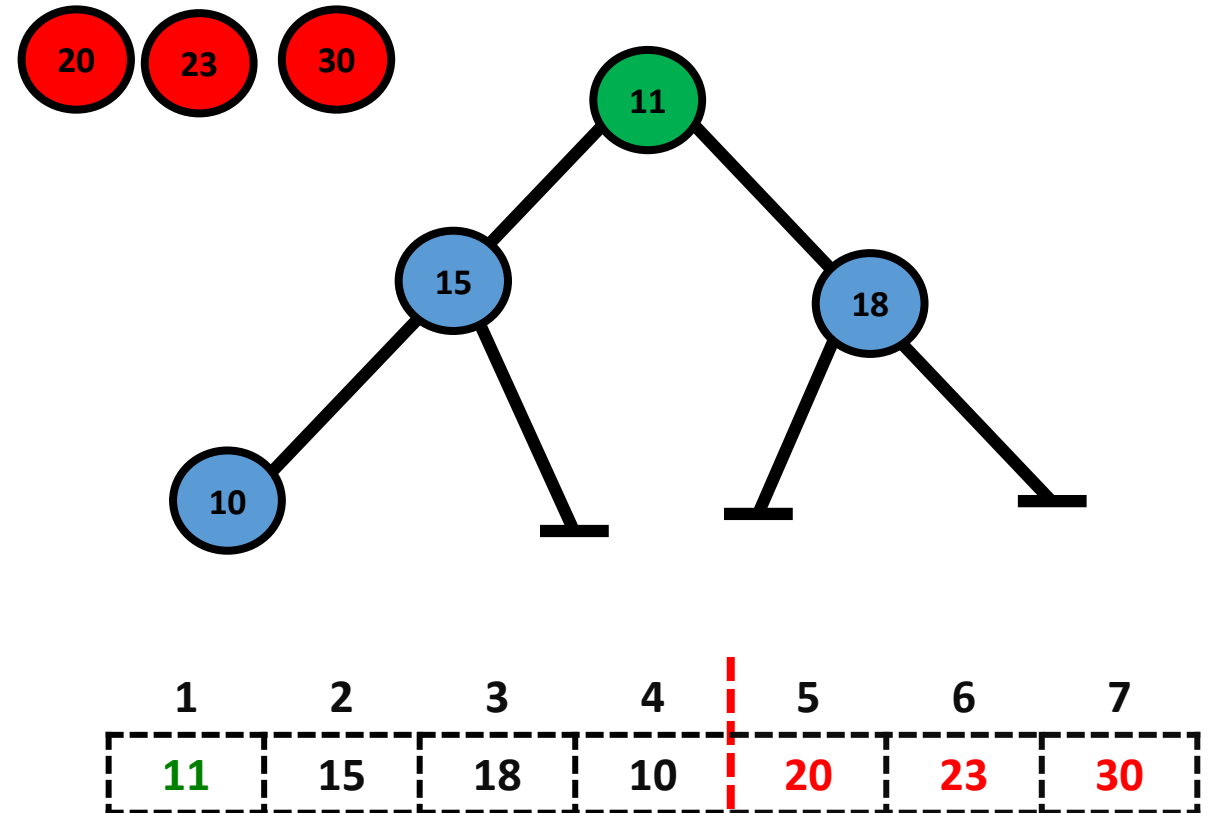


Ștergerea elementelor din Heap (Max)

Max Heap

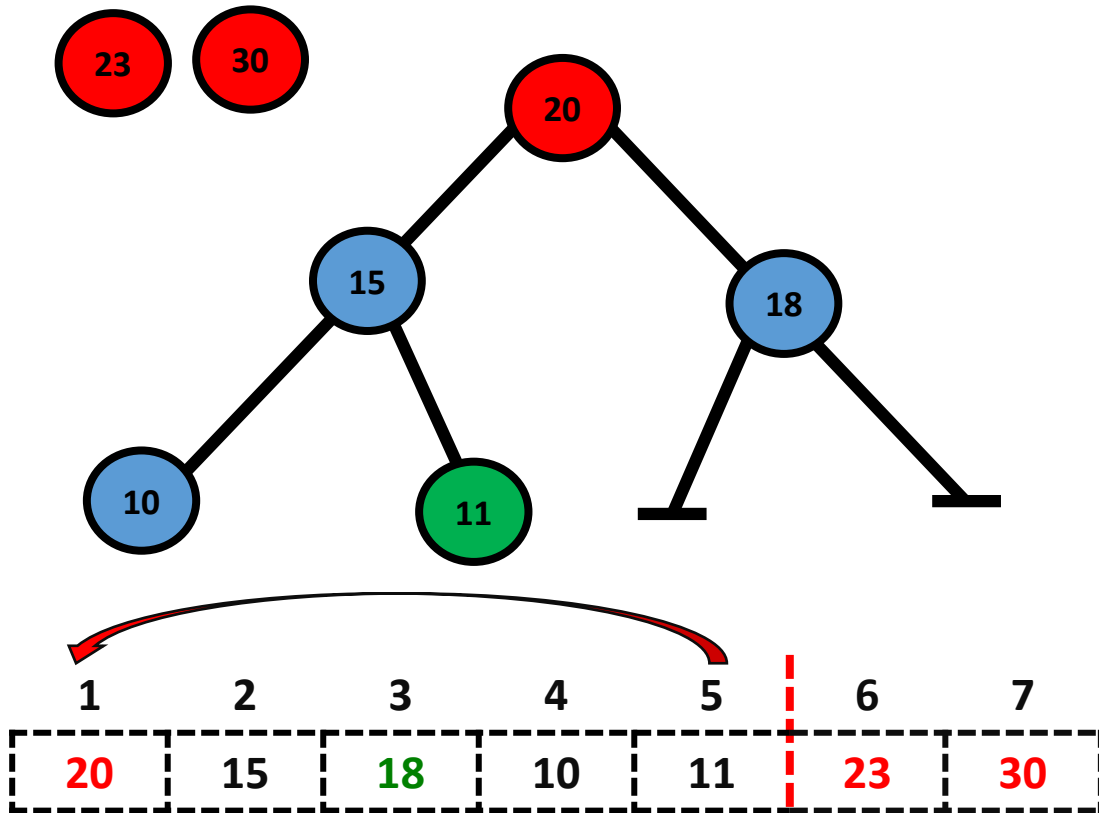


Max Heap

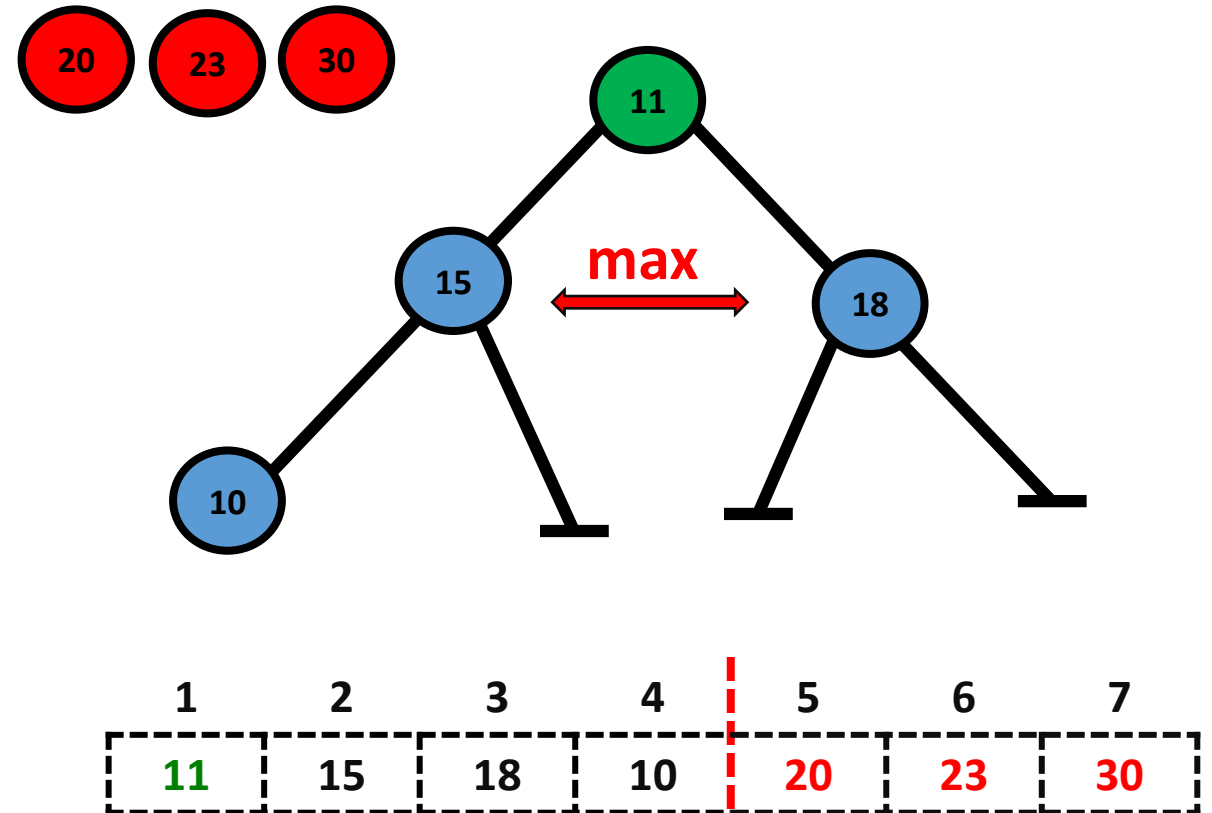


Ștergerea elementelor din Heap (Max)

Max Heap

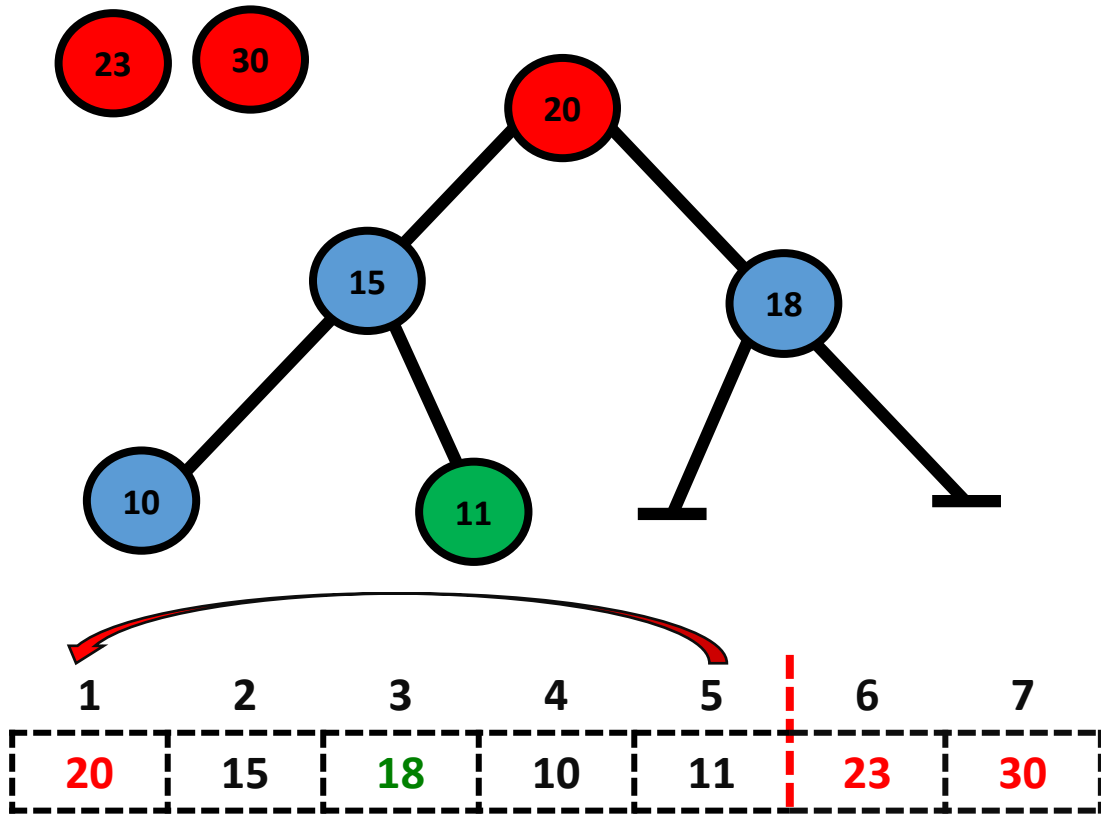


Max Heap

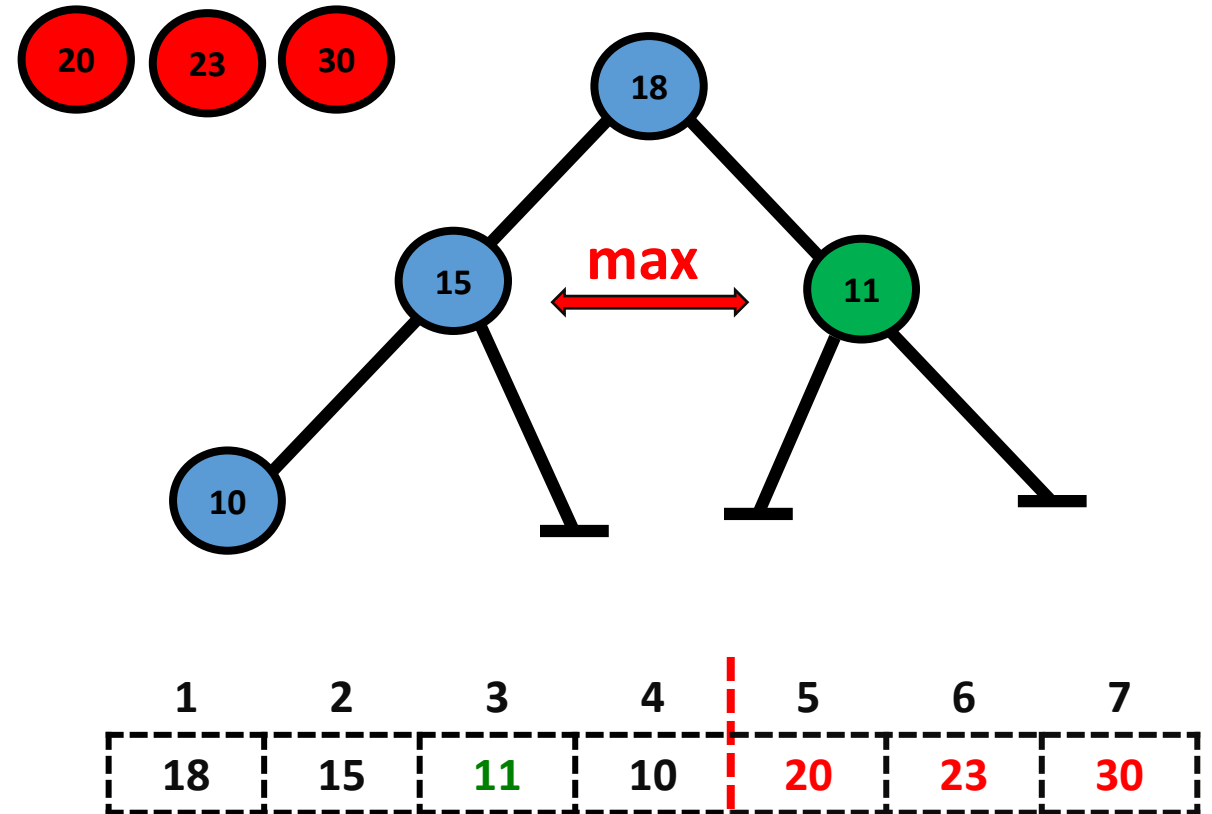


Ștergerea elementelor din Heap (Max)

Max Heap

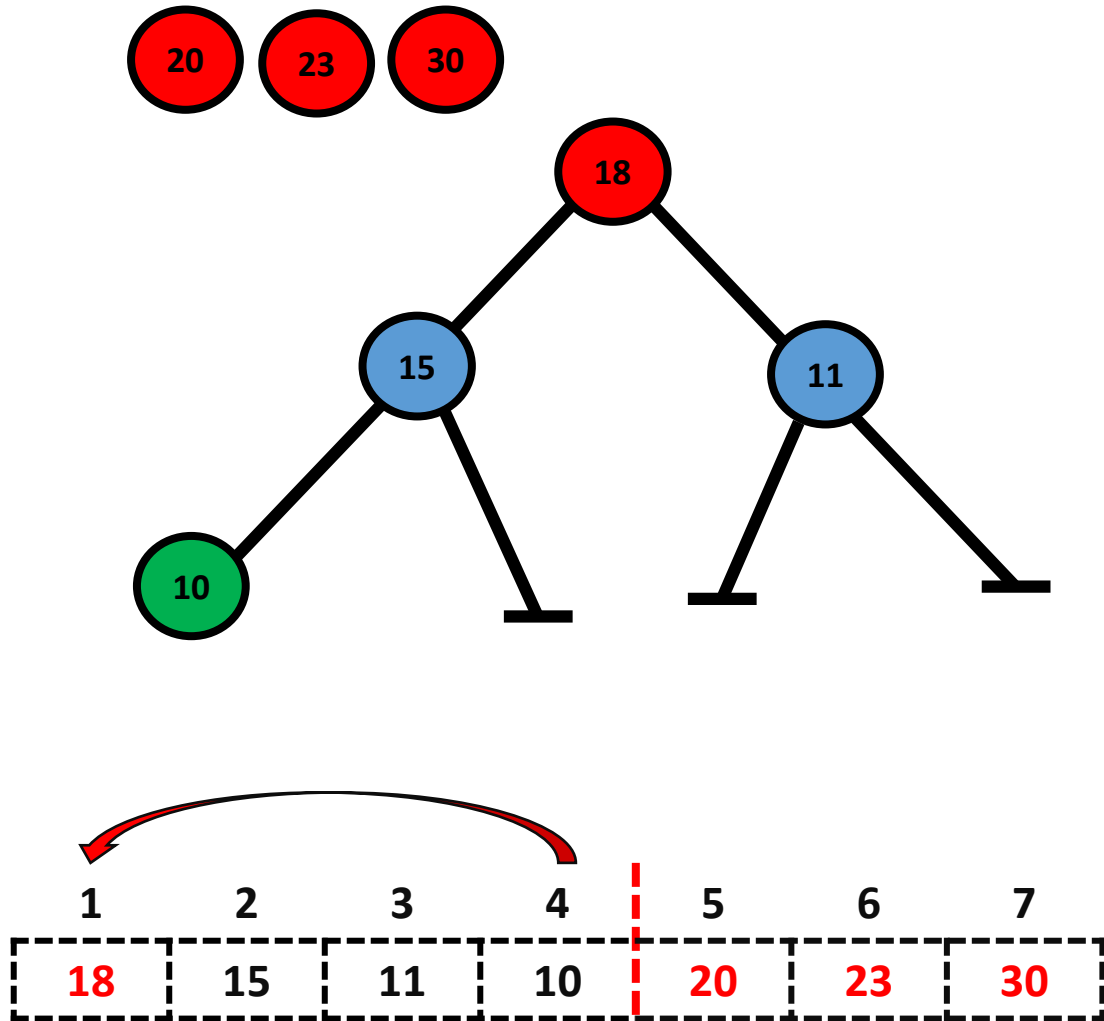


Max Heap

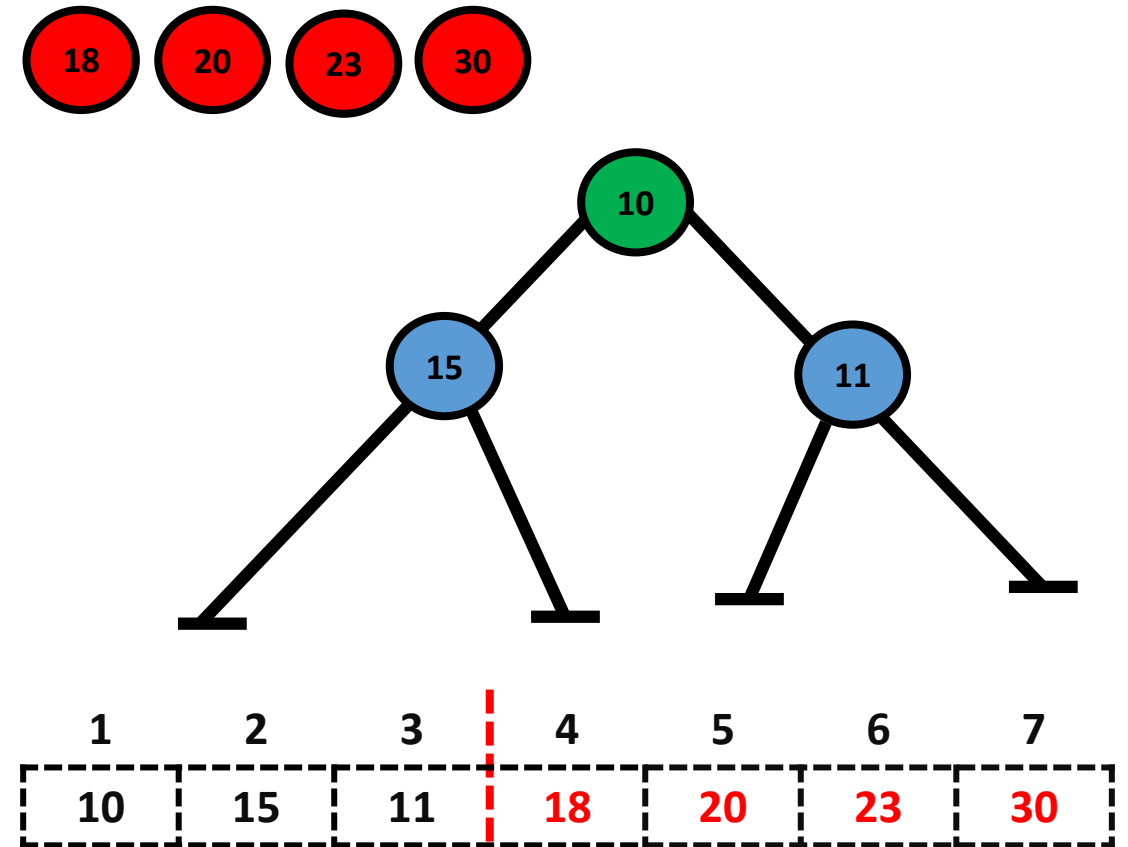


Ștergerea elementelor din Heap (Max)

Max Heap



Max Heap



!!! Elementele șterse din heap și rescrise în array, devin aranjate crescător.

Heapsort

Heapify

Priority Queues

- <https://www.youtube.com/watch?v=HqPJF2L5h9U>

Nodul	Indexul 1	Indexul zero
Un nod	i	i
Copilul din stânga a nodului	$2*i$	$2*i+1$
Copilul din dreapta a nodului	$2*i+1$	$2*i+2$
Părintele nodului	$i/2$	$(i-1)/2$