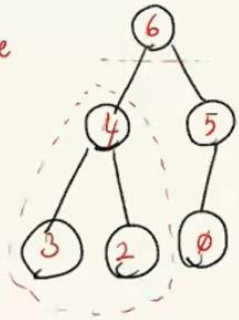
19.04.2021
Week 7

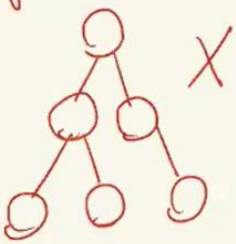
Heap Tree

→ Complete Binary Tree

Max-Heap Tree

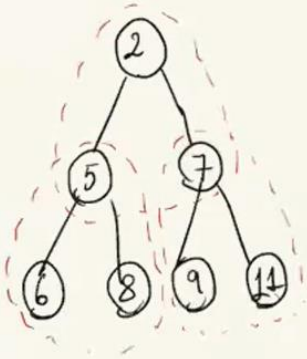


✓



X

Min-Heap Tree

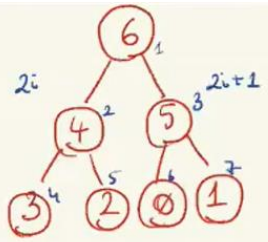


Max Heap : $A[parent(i)] \geq A[i]$

Min Heap : $A[parent(i)] \leq A[i]$

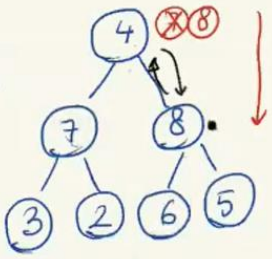
Array :

☐	6	4	5	3	2	0	1
Ø							




```
void max_heapify(int dizi[], int i, int N){
    int left= 2*i;
    int right= 2*i+1;
    if (left<=N && dizi[left]>dizi[i])
        Largest=left;
    else
        largest = i;
    if (right<=N && dizi[right]>dizi[largest])
        Largest=right;
    if (largest!=i){
        swap(dizi[i], dizi[largest]);
        max_heapify(dizi, Largest, N);
    }
}
```

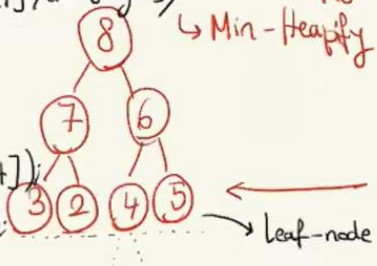
Max-Heap:



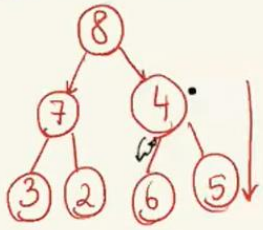
Heapify Operation

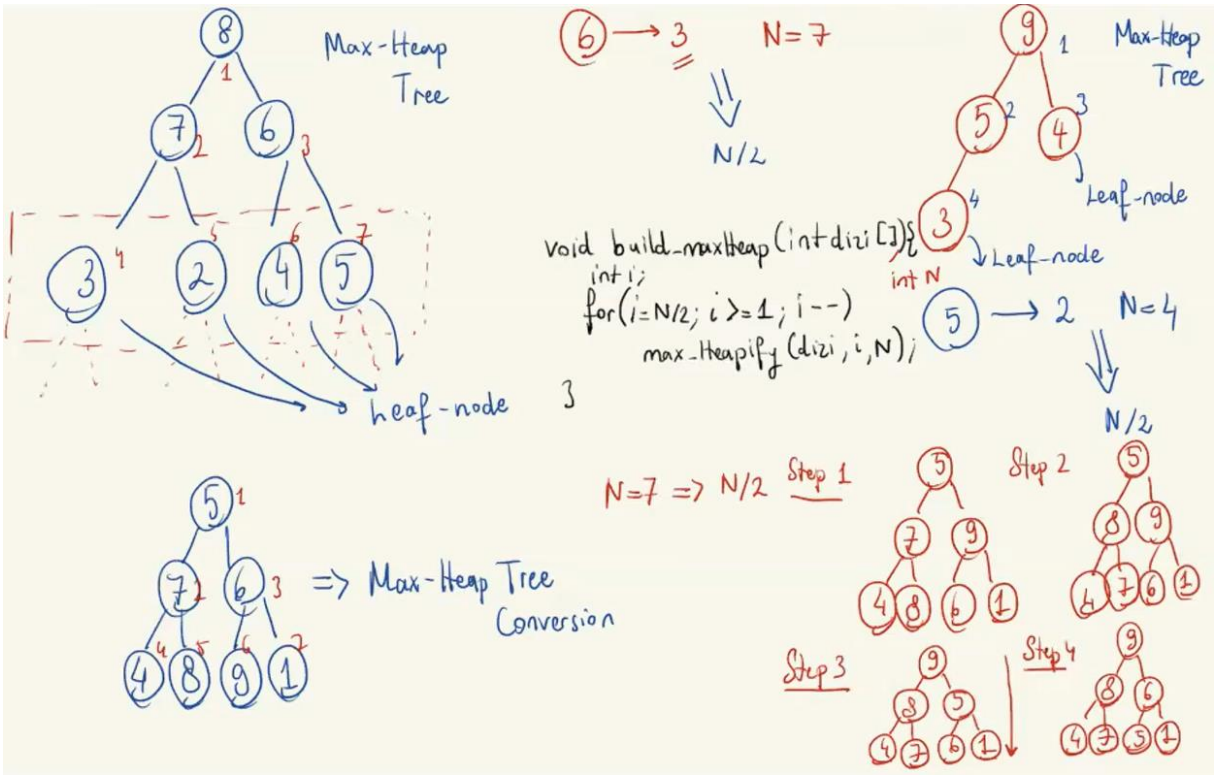
↳ Max-Heapify

↳ Min-Heapify



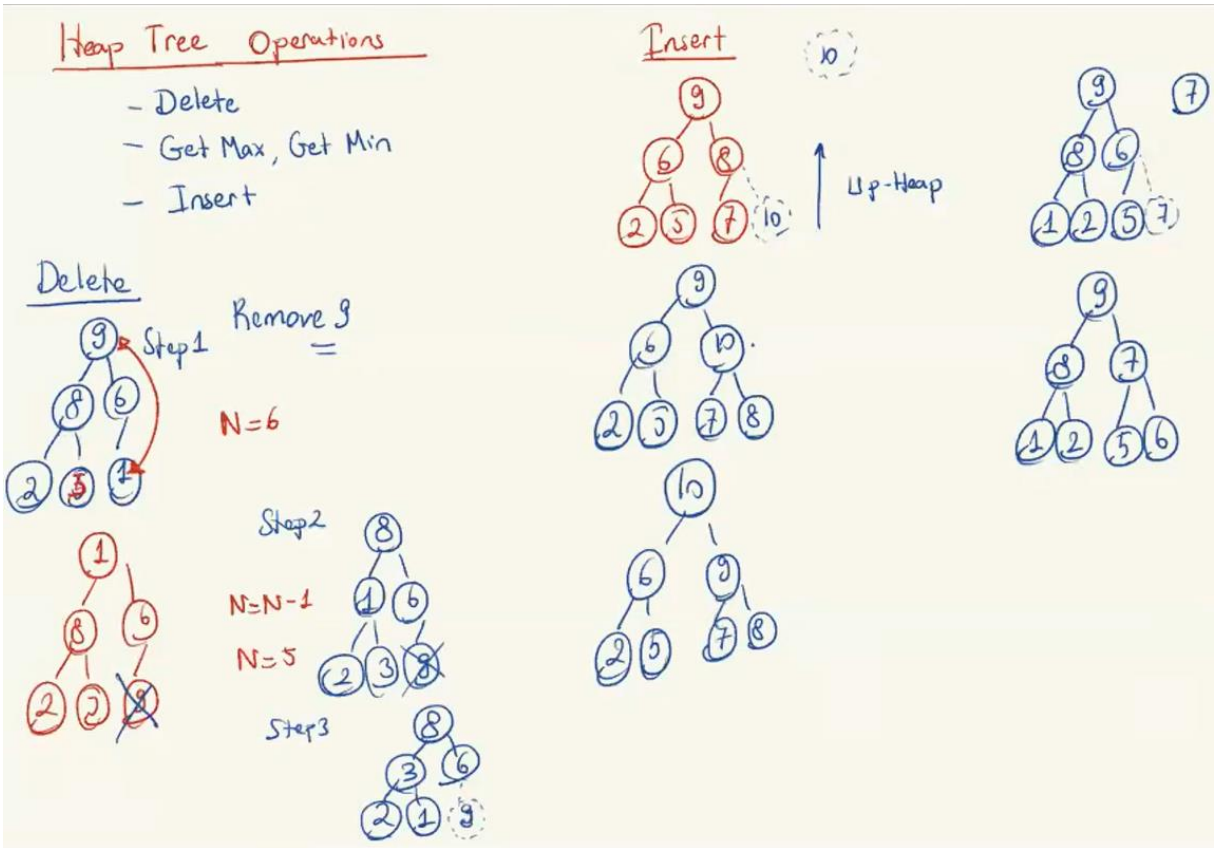
→ Leaf-node

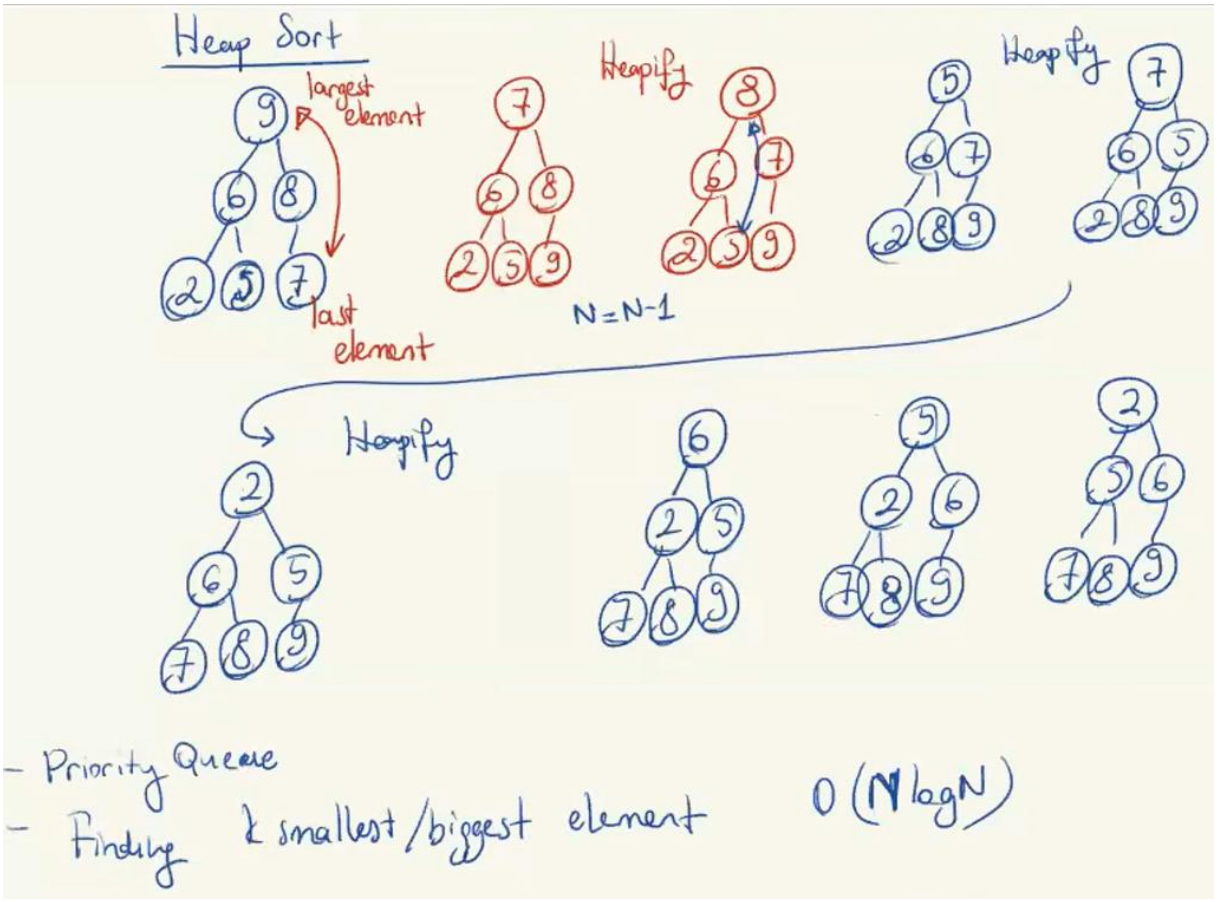




Heap Tree Operations

- Delete
- Get Max, Get Min
- Insert





- Priority Queue
- Finding k smallest / biggest element $O(N \log N)$