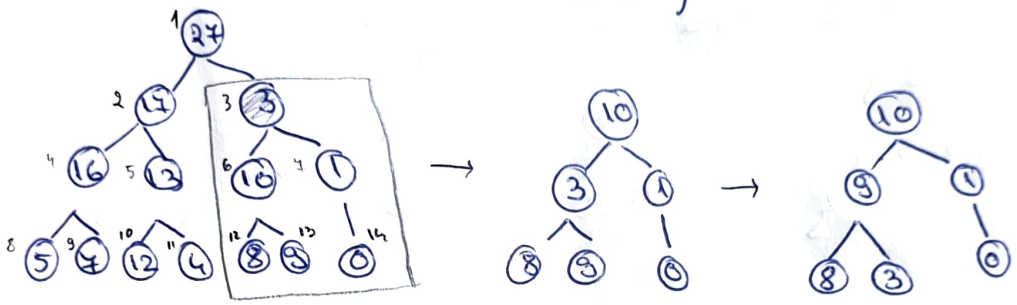
Struct. Date

06.03.2024

Seminar 1

① (MAX-)HEAPIFY (A, 3) → poziția elem. de unde mă interesează ordonarea

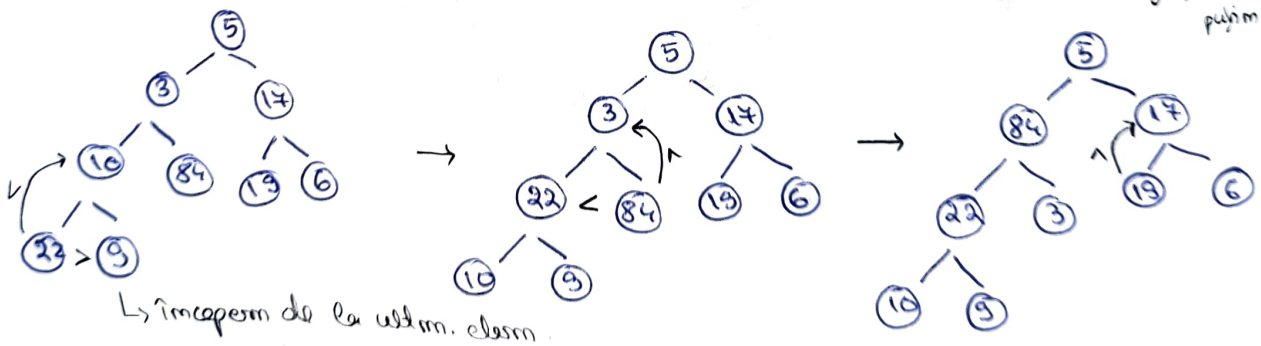
A = {27, 17, 3, 16, 13, 10, 1, 5, 7, 12, 4, 8, 9, 0}



Vectorul după heapify: A = {27, 17, 10, 16, 13, 9, 1, 5, 7, 12, 4, 8, 3, 0}

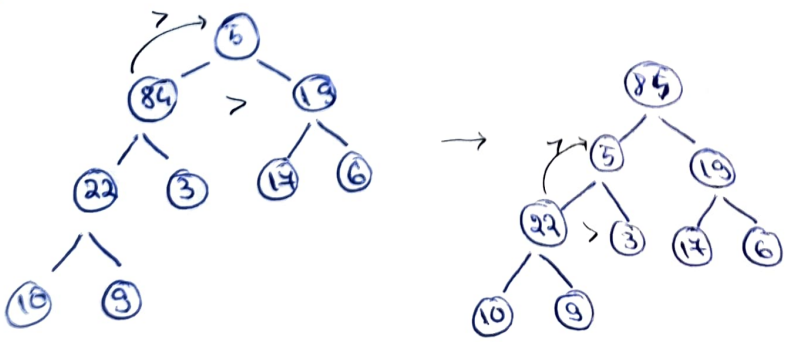
② (MAX)BUILD => heapify invers

A = {5, 3, 17, 10, 84, 19, 6, 22, 9} ← parcurgi vectorul inv. (în schemă am grupat ord. pt. a fi mai ușor)

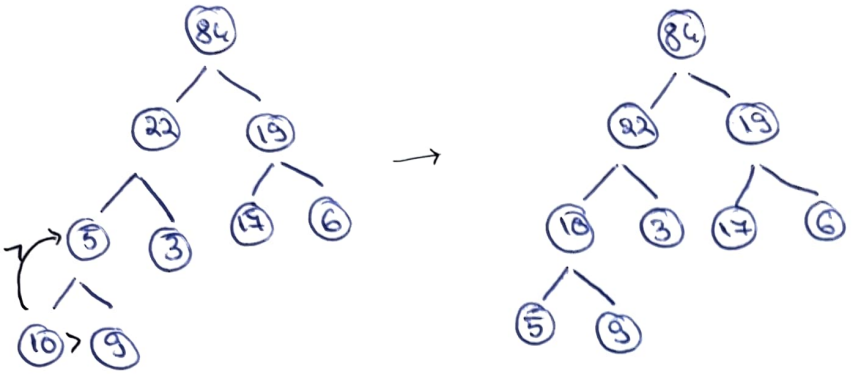


↳ începem de la ultim. elem. comparăm cu cel de lângă el

Pt că (14) are copii, atunci aplicăm heapify pe copiii lui și după comutăm



Acum problema e că 5 < 22, așa că aplicăm iar heapify pt copiii lui 5



Vectorul după max build : $A = \{84, 22, 19, 10, 3, 14, 6, 5, 9\}$

③ MERGE SORT pt k vectori cu $O(n \log k)$. Exemplu pe $k = 3$

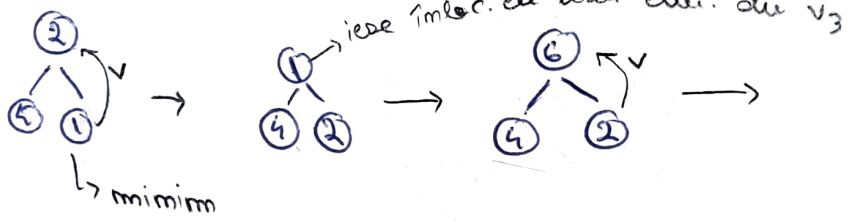
$V_1 = [2, 3, 8]$

$V_2 = [4, 5, 9]$

$V_3 = [1, 6, 7, 11]$

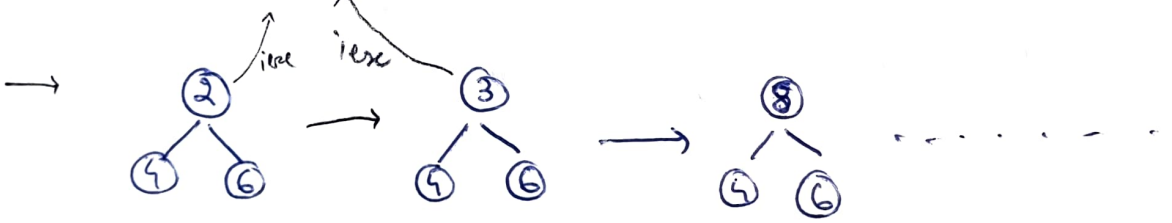
$\vdots$

$V_k =$



Aplicăm heapify pe fiecare „col", pe k elem

Vector = [1, 2, 3, ...



Pt k elem / vectori =>

$V_1 = \{a_1, a_2, \dots a_j\}$

$V_2 = \{a_{j+1}, \dots a_i\}$

$\vdots$

$V_k = [a_{l+i}, \dots a_m]$