CSE 101 3-10-26

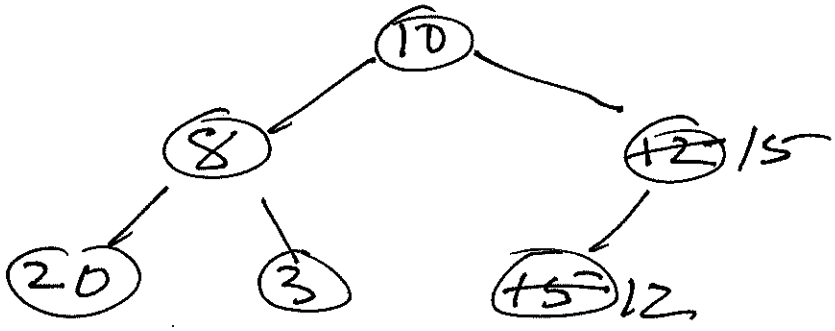
Pset: ext 2 days (last)

6.4 HeapSort

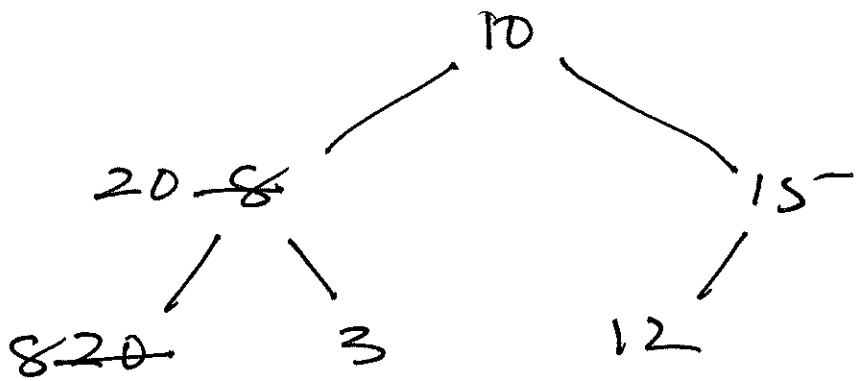
Ex. $A = (10, 8, 12, 20, 3, 15)$

First: BuildHeap:

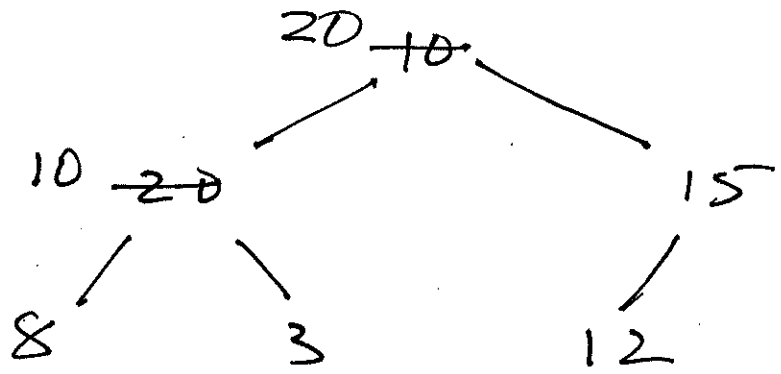
1	2	3	4	5	6
10	8	~~12~~ 15	20	3	~~15~~ 12



A | 10 | ~~8~~ | 15 | ~~20~~ | 3 | 12 |
20 8

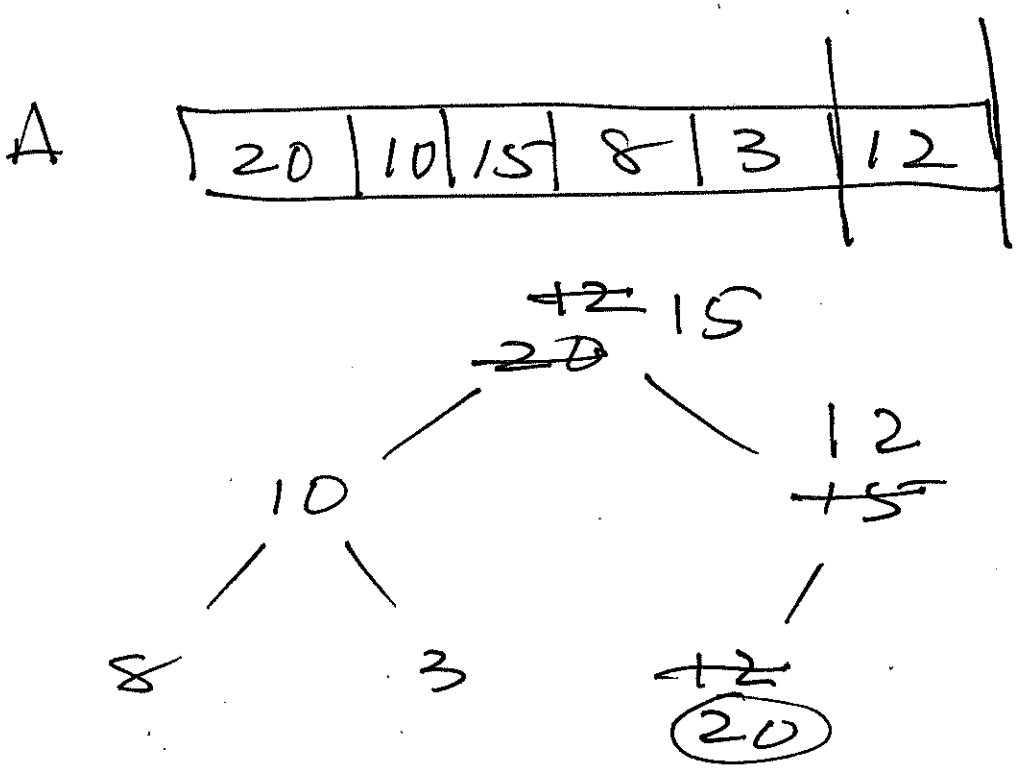


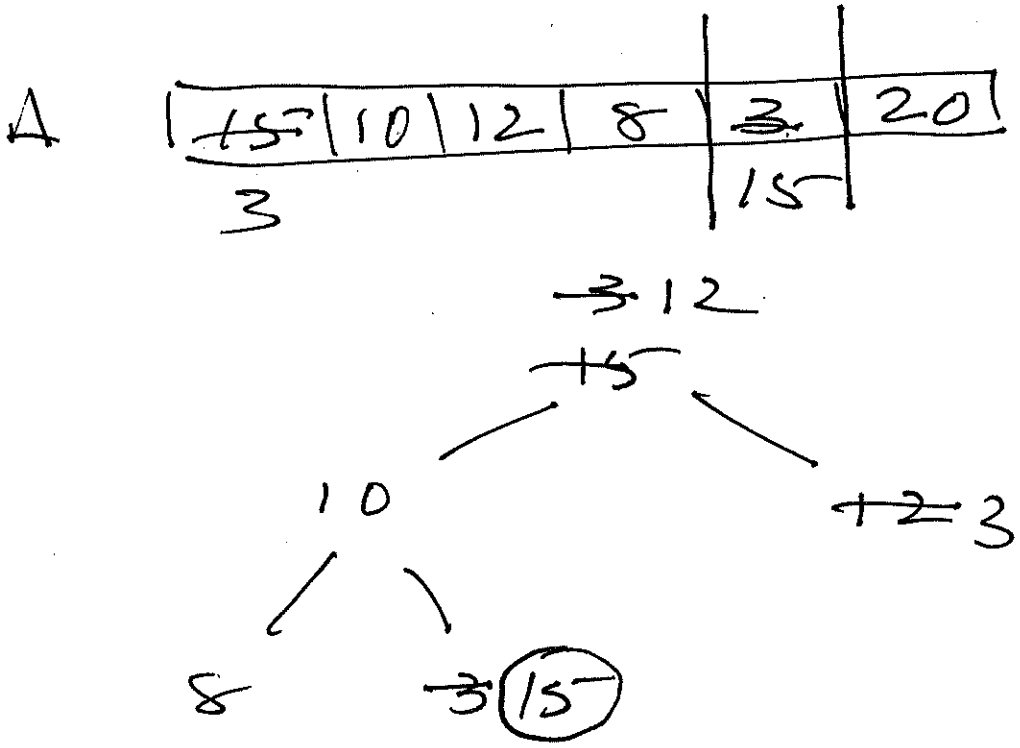
A | 10 | 20 | 15 | 8 | 3 | 12 |

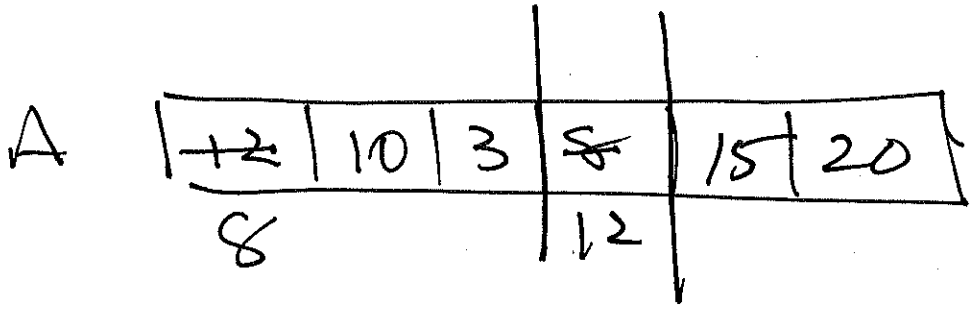


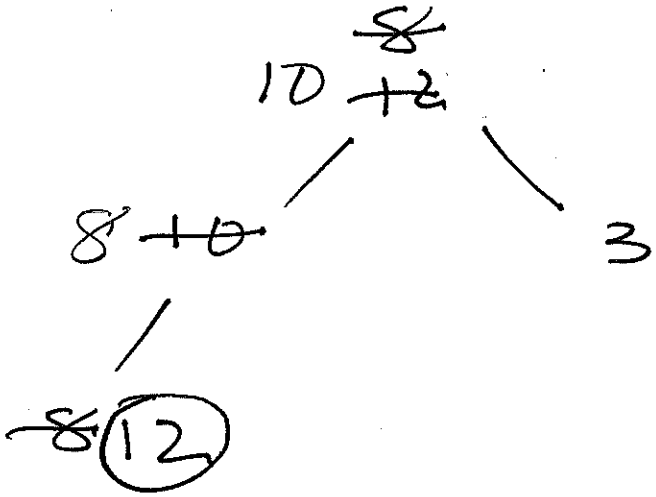
| 1 | 2 | 3 | 4 | 5 | 6 |
A | 20 | 10 | 15 | 8 | 3 | 12 |

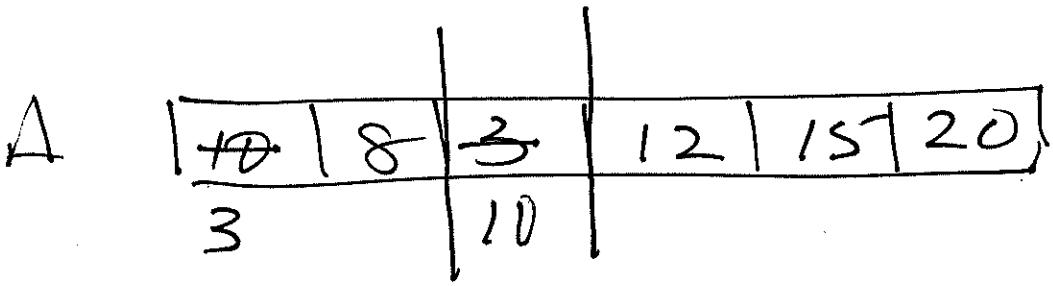
Rest of Heapsort:

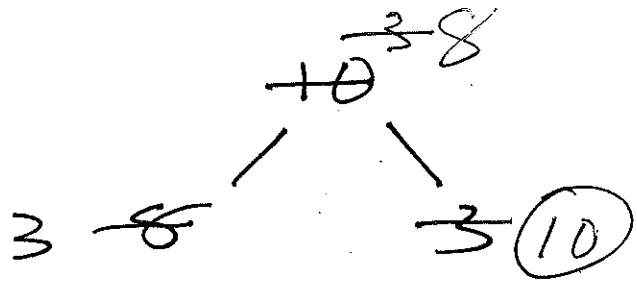


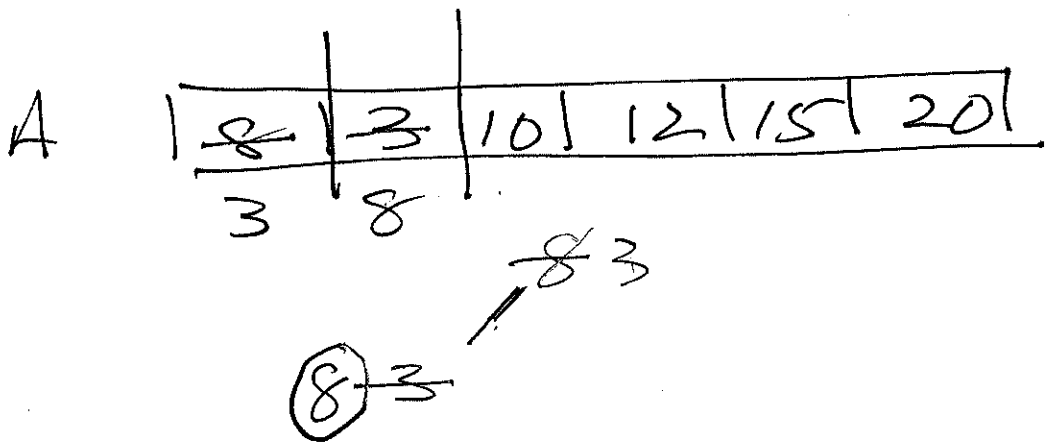



A
12
10
3
8
15
20
8
12


8
10
12
8
10
3
8
12


A
10
8
3
12
15
20
3
10


3
8
10
3
8
3
10


A
8
3
10
12
15
20
3
8
8
3
8
3

A

3	8	10	12	15	20

Runtime: $\Theta(n \log n)$

PQ OPS

HeapMax(): $\Theta(1)$

HeapDeleteMax(): $\Theta(\log n)$

HeapExtractMax(): $\Theta(\log n)$

HeapIncreaseKey(): $\Theta(\log n)$

HeapInsert(): $\Theta(\log n)$

24.1 - 24.4 Weighted graphs

Defn

- A **weighted graph** $G = (V, E, w)$ is a graph (V, E) and a function

$$w : E \to \mathbb{R}$$

- let p be an x-y path in G

$$p : x = v_0, v_1, \ldots, v_k = y$$

Then the weight of p is

$$w(p) = \sum_{i=0}^{k-1} w(v_i, v_{i+1})$$

- Let $x, y \in V(G)$. The <u>shortest path weight</u> $\delta(x,y)$ is

$$\delta(x,y) = \begin{cases} \min\{w(p) \mid p \text{ is an } x\text{-}y \text{ path}\} & \text{if } y \text{ is reachable from } x \\ \infty & \text{otherwise} \end{cases}$$

- a <u>shortest x-y path</u> p is an x-y path p such that

$$w(p) = \delta(x,y).$$

- SSSP Problem

Given a Weighted Graph G, and a source $s \in V(G)$

(a) determine $\delta(s,x)$ for all $x \in V(G)$

(b) For those $x \in V(G)$ for which $\delta(s,x) < \infty$, also determine a shortest $s-x$ Path P.

Algorithms

- Dijkstra
- Bellman-Ford

Both algorithms actually find min. weight walks.

Problem: negative weight cycles reachable from source s.