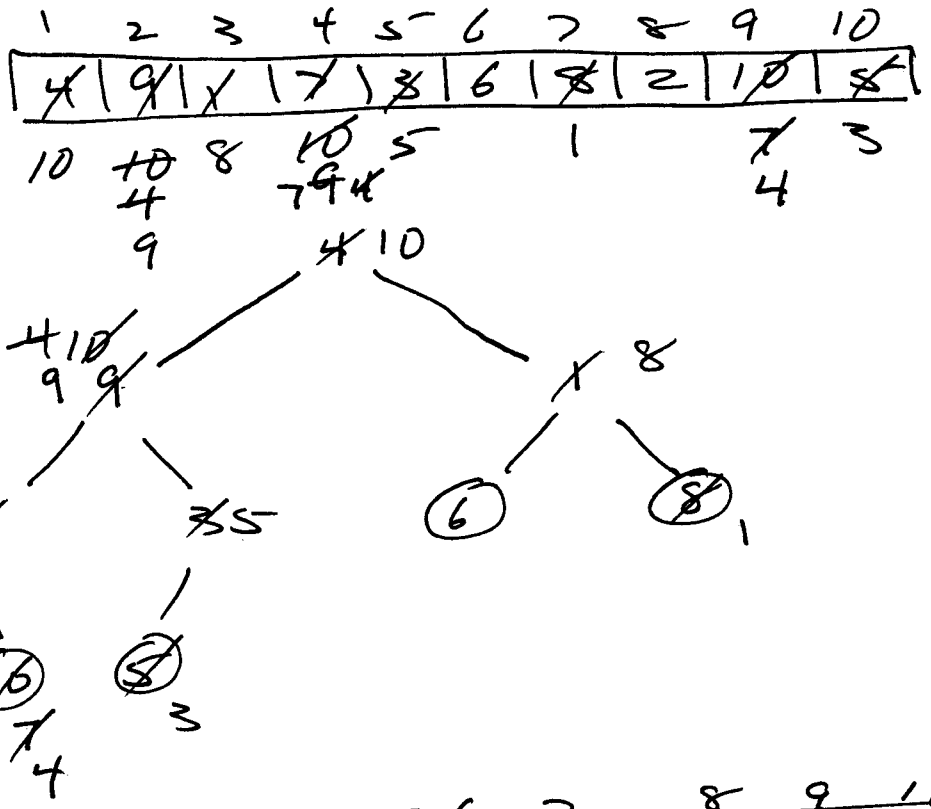
CSE 101 2-25-21

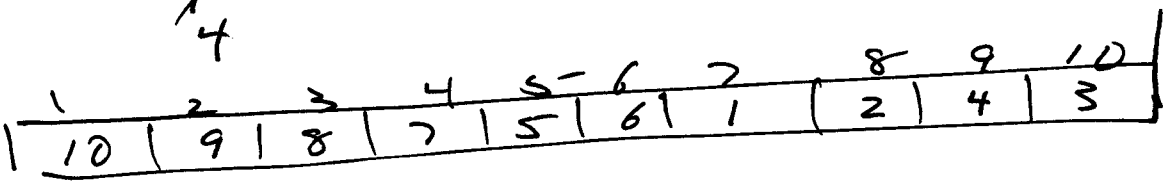
Pa 6: ext 2 more days (no more!!)

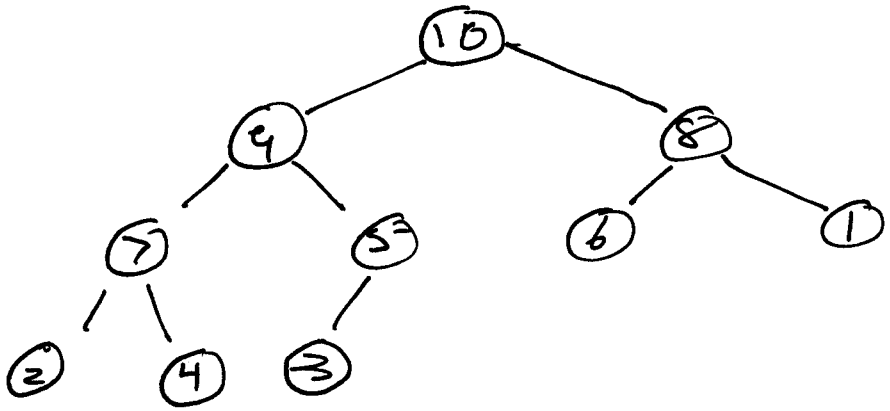
BuildHeap

Ex. A



A





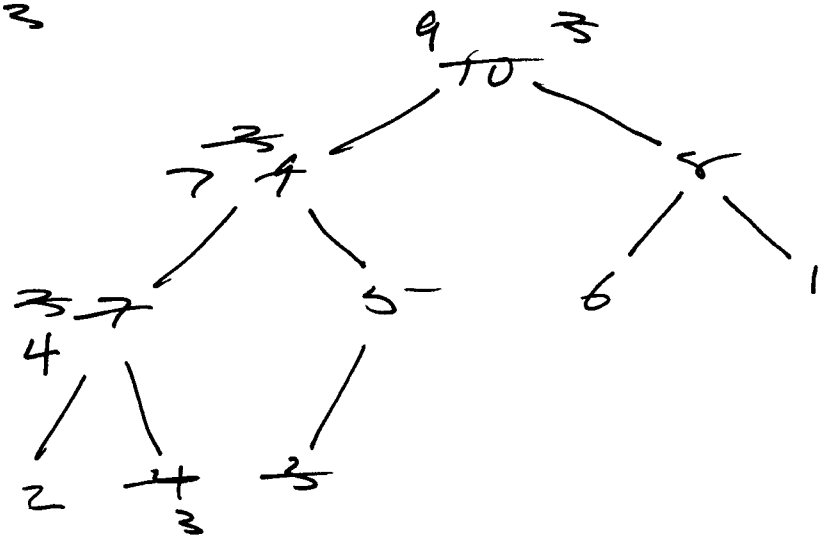
Runtime of BuildHeap :

$\Theta(n)$.

6.4 HeapSort : Same example.

	1	2	3	4	5	6	7	8	9	10
A	10	9	8	7	5	6	1	2	4	3
										10

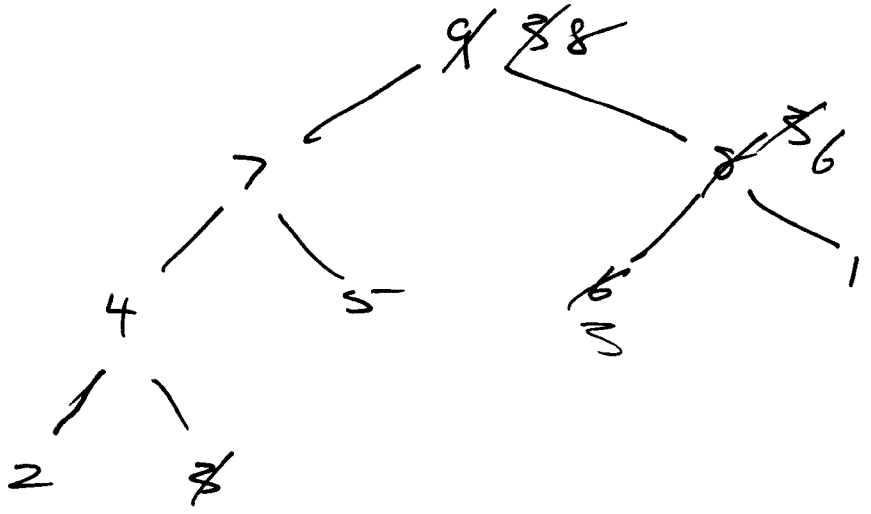
3



A

1	2	3	4	5	6	7	8	9	10
~~9~~	7	~~8~~	4	5	6	1	2	~~3~~	10

~~3~~ 9



A

1	2	3	4	5	6	7	8	9	10
8	7	6	4	5	3	1	2	9	10

Exercise continue this.

Runtime $= (n-1) \cdot \Theta(\log n)$

$= \Theta(n \log n)$.

note: $\log_2(n!) = \Theta(n \log n)$

6.5 Priority Queues

A max Priority Queue is an ADT that maintains a set S of records, each with a field called key.

$$x = (\cdot, \cdot, \cdot, key) \in S$$

$$S = \{\cdot, \cdot, \cdot, x, \cdot, \cdot\}$$

- a state is a particular set S.
- Operations.
  - Insert(S, x): adds new record
  - Max(S): returns record w/ max key.
  - ExtractMax(S): returns and deletes
  - DeleteMax(S): deletes rec. w/ max ke
  - ~~IncreaseKey(S, x,~~

IncreaseKey(S, x, k):

change key[x] to k > key[x]

Implementing max P.Q. as max heap:

see pseudo-code on webpage.

Runtimes:

HeapMaximum() : $\Theta(1)$

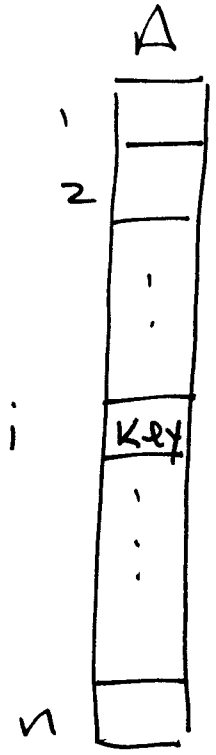
HeapDeleteMax() : $\Theta(\log n)$

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

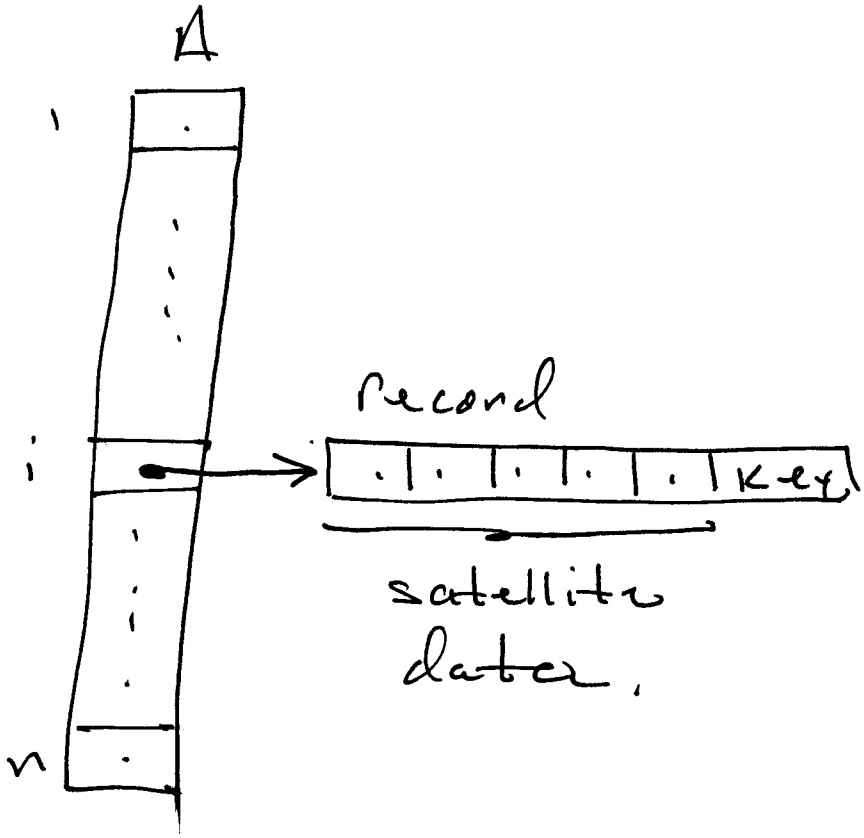
HeapIncreaseKey() : $\Theta(\log n)$

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

our Picture



general Picture



Exercise: re-write P.Q. algorithms in the general Picture

max heap Property:

$$key[A[i]] \leq key[A[Parent(i)]]$$

Exercise

re-write everything for a min-heap and min-Priority Queue

Exercise

Implement the P.Q. ADT (both min & max) in both C and C++.

Next time: chap 24

- weighted graphs
- Path weight
- min Path weight
- shortest (lightest) Path
- SSSP in a weighted graph.

Quiz #4 Prob. 1e

Keys: 5, 9, 7, 2, 6, 4, 8, 3, 1, 10