CS2100

Priority Queues

Heaps

Recap

- No class next week
- HW due Sunday after break
- Lab due today

No Office hours today

Last time:

Priority queues: store 2 things:
- value
- key

Operations: Given priority queue PQ:

- insert(e, k): adds e to PQ w/ priority k
- getMax(): returns maximum ~~value~~ key in PQ
- removeMax():

(plus size & empty)

Note: Many possible implemtations!

We talked about 2 using vectors.

One (best?) way: heaps

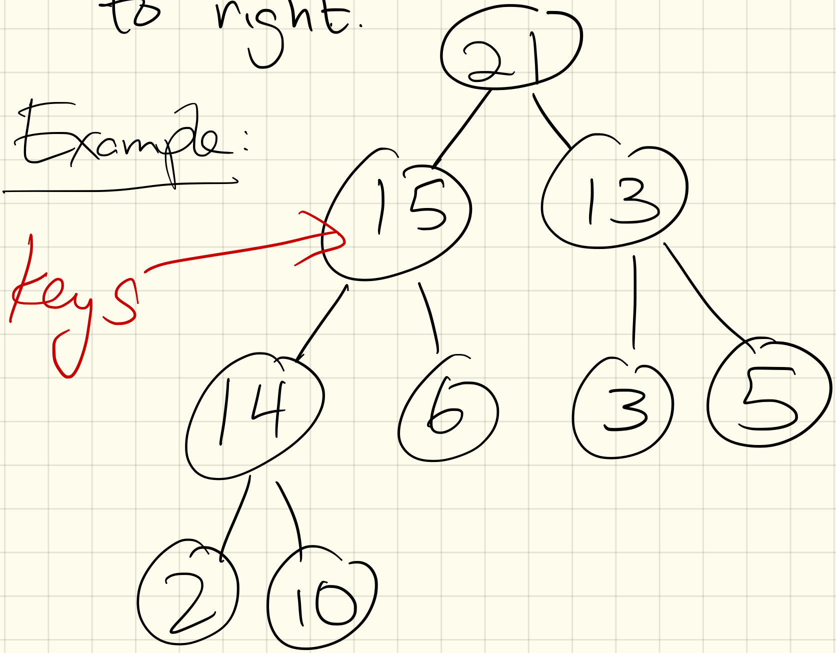
Heap: A binary tree where:

One way to do a priority queue

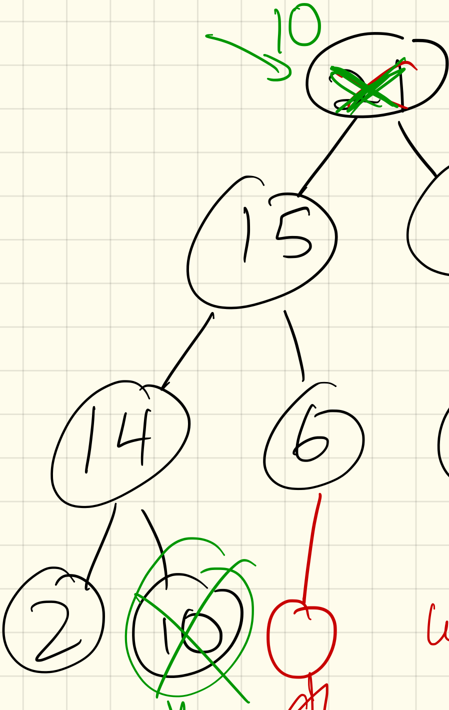
#1
- For every node v (other than r) the Key stored at v is ≤ ~~≥~~ key stored at v's parent

#2
- The tree is complete: levels 0... h-1 are full + h is filled in left to right.

Example:



3 functions: insert, getMax, $O(1)$ removeMax.



while (curr < either child)
swap w/ larger child

next insert must go here

while (curr > parent)
move up

Now: Code! (on webpage)

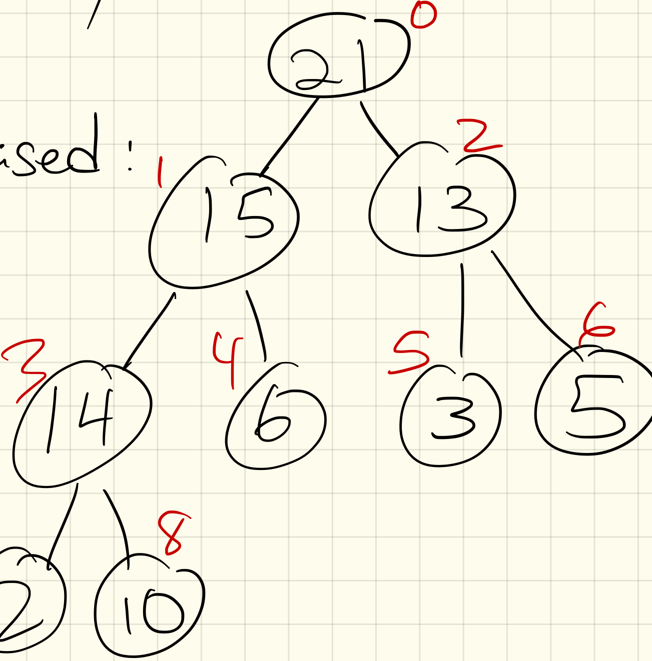
Recall: Array based trees:

- Array based:

left(v) = $2v+1$

right(v) = $2v+2$

parent(v) = $\lceil \frac{v}{2} \rceil - 1$



A-	21	15	13	14	6	3	5	2	10						
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Runtimes:

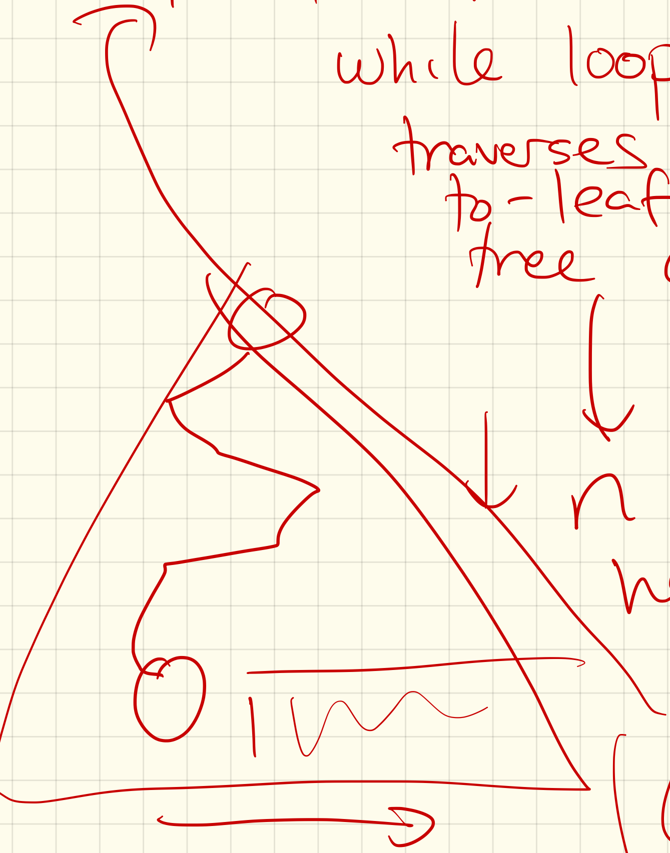
empty + size : $O(1)$

getMax : $O(1)$

insert + remove:

while loop:

traverses a root-to-leaf path in the tree once



n nodes

height $\lceil \log_2 n \rceil + 1$

$$\boxed{O(\log_2 n)}$$