

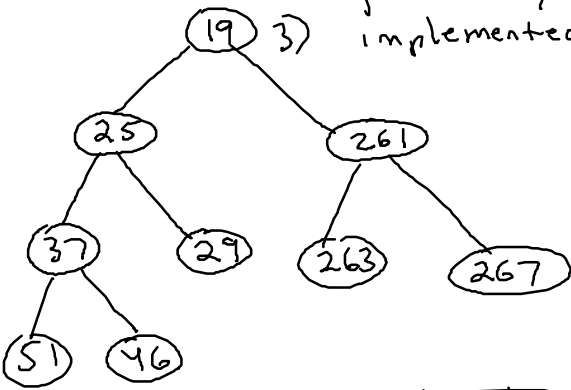
CS 225

Priority Queues : Implementation

Disjoint Sets : ADT

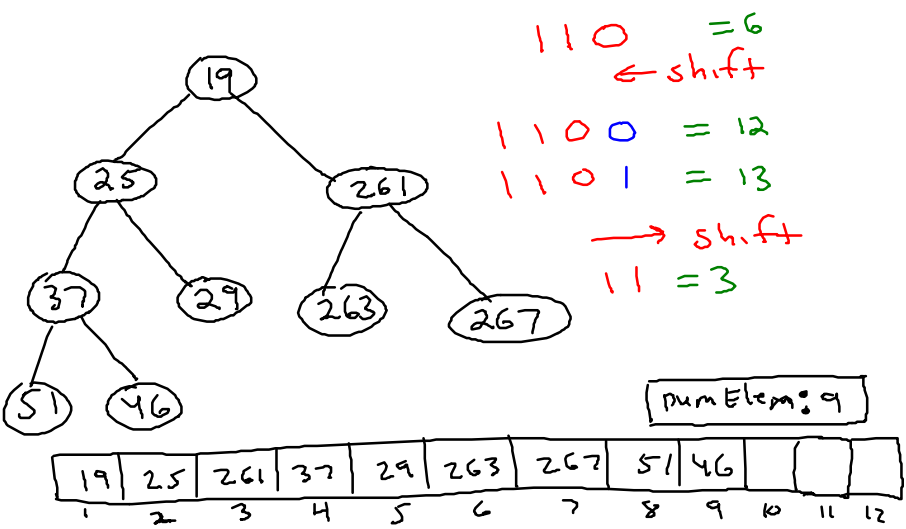
heap :

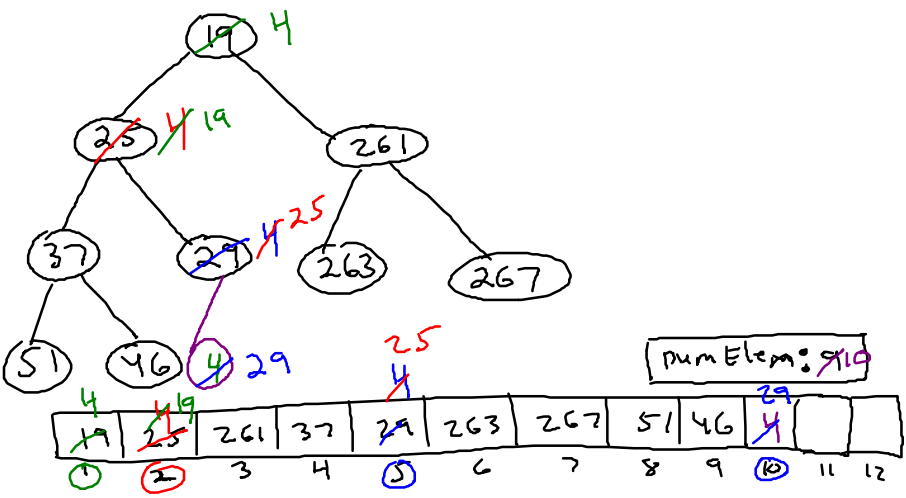
- 1) complete tree
- 2) partially ordered
- 3) implemented via array

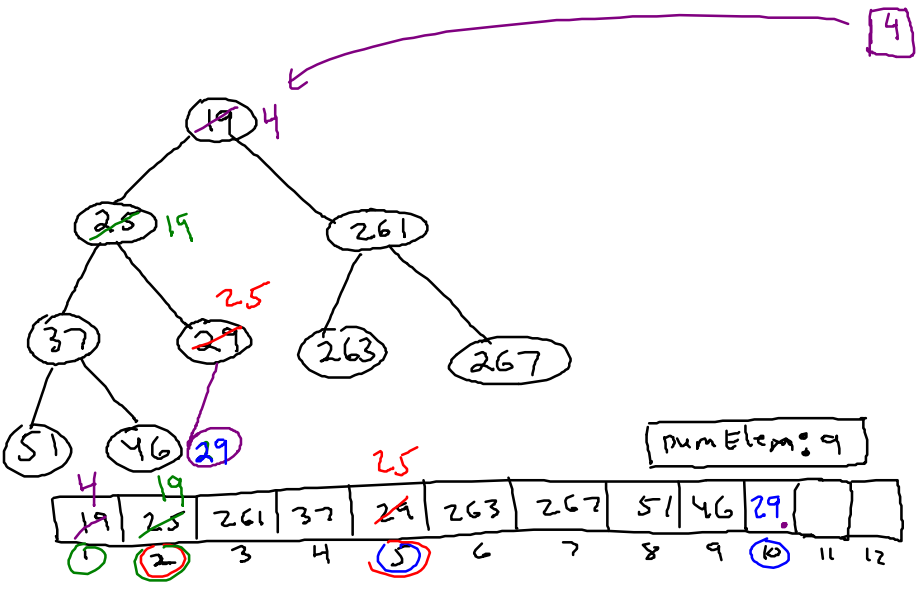


numElem: 9

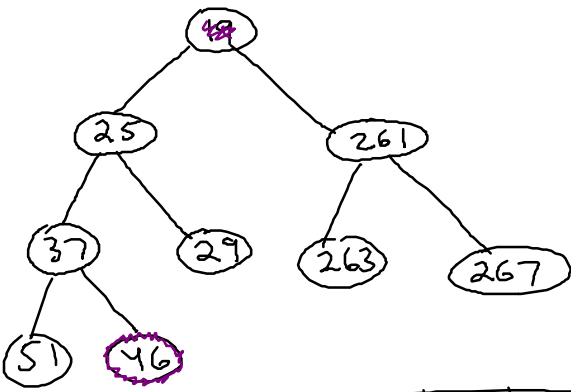
19	25	261	37	29	263	267	51	46			
1	2	3	4	5	6	7	8	9	10	11	12







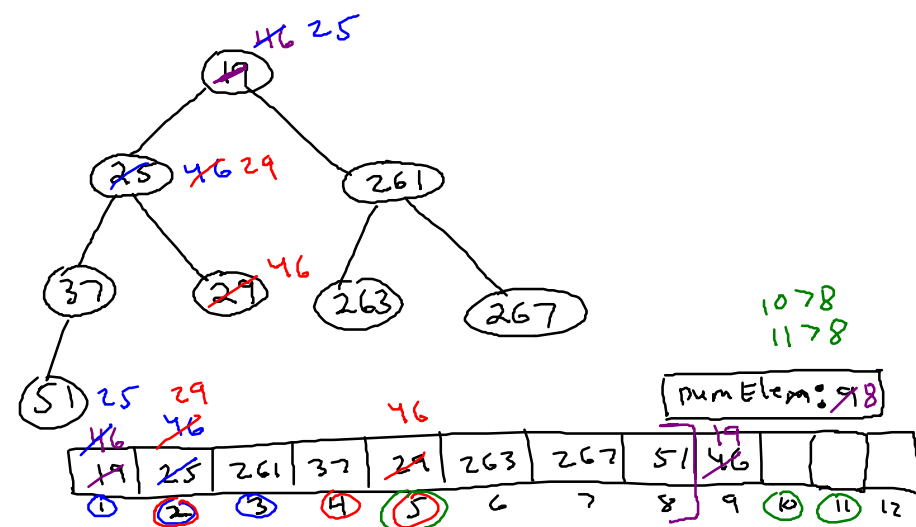
19



numElem: 9

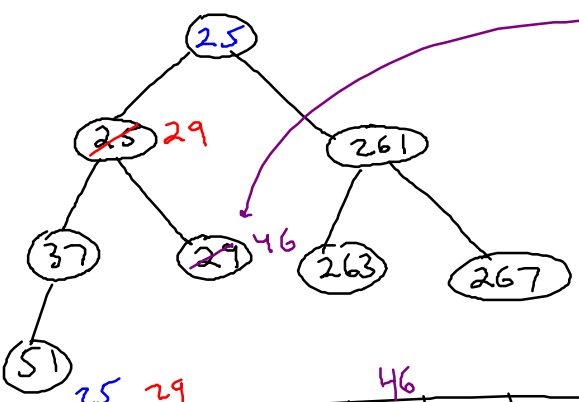
19	25	261	37	29	263	267	51	46			
1	2	3	4	5	6	7	8	9	10	11	12

119



19

46



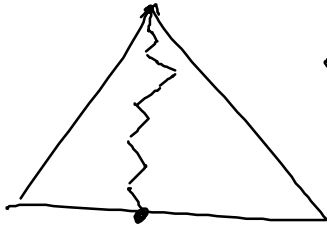
numElem: 9

25	29			46								
19	25	261	37	29	263	267	51	46				
1	2	3	4	5	6	7	8	9	10	11	12	

Find Min : $O(1)$
isEmpty : $O(1)$

Insert

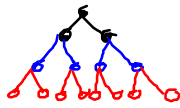
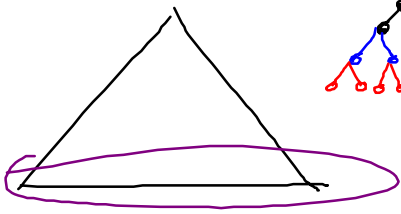
↳ worst case



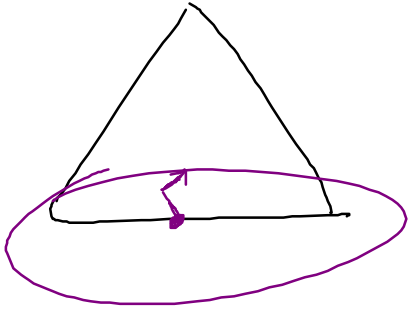
• constant time per level

• $O(\lg n)$ levels

↳ insert in worst case is $O(\lg n)$



perfect tree, $> 50\%$
of values are in
bottom row of
tree



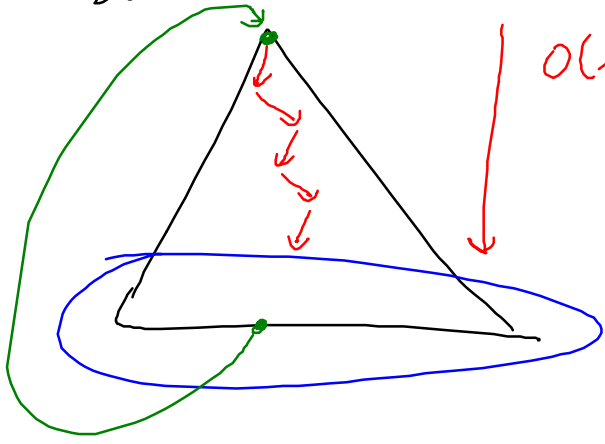
complete tree, $> 50\%$
of values are in
bottom two rows

insert avg case

↳ value inserted moves
up two or three levels

⇒ $O(1)$ insert
avg
case

Delete Min



$O(\lg n)$

Delete Min
avg case &
worst case
are $O(\lg n)$

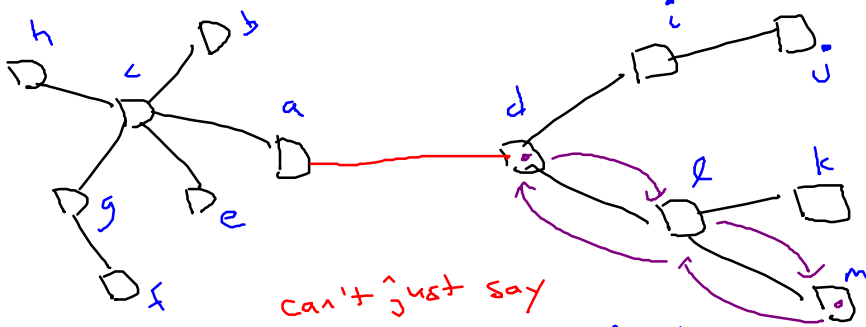
Disjoint Sets

$\equiv 0$	$\equiv 3$
$\equiv 1$	$\equiv 4$
$\equiv 2$	$\equiv 5$

$$1) S_1 \cup S_2 \cup \dots \cup S_n = U$$

$$2) S_i \cap S_j = \emptyset$$

$i \neq j$



can't just say

$a R c$
 $c R b$
 $a R b$
 etc

\Rightarrow $a R d$
 $d R a$
 $a R c$
 $a R b$
 $d R c$
 etc

$l R d$
 $d R l$
 $l R i$
 $d R i$
 $i R j$
 $l R j$
 $j R l$
 etc

void Union (Set A, Set B)

to add aRd

what set is "a" in?

what set is "d" in

if not same set, union sets

Set Find (element x)

to find out if aRd

$\text{Find}(a) == \text{Find}(d)$