Final Exam CS 302

Name: Alan Reisenauer
NSHE ID: 2001751827

Question 1:

What's the time complexity of the below operations:

```
(a). for (let i = 0; i < n; i ++){ — O(n)
          for (let j = 0; j < m; j++){ -O(m)
                    cout << i << " " << j;}}
```

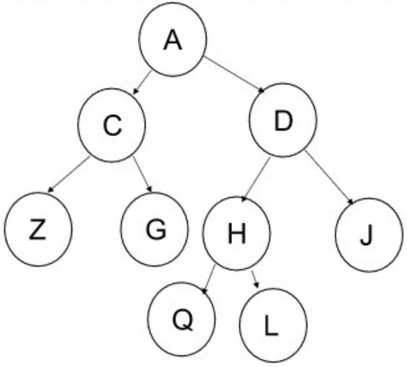
$O(m \cdot n)$

```
(b). for (int i = n; i > 0; i=i/2){
          for (int j = 0; j< n; j++){
                    cout << “hello”; }}
```

$O(n)$

Question 2:

What's the pre-order, in-order traversal of the below tree?



Pre order: A, C, Z, G, D, H, Q, L, J

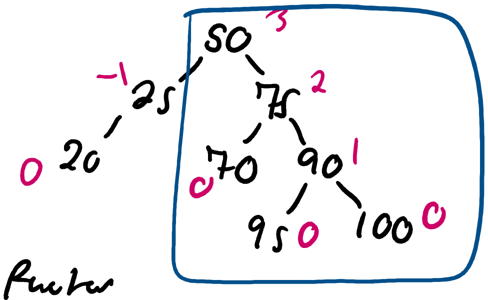
In order: Z, C, G, A, Q, H, L, J, D

L, R, Ro Post Order: Z, G, C, Q, L, H, J, D, A

Question 3:

Insert below into a AVL tree. Please write down each step of rotation to make it an AVL tree.
Insert: 50, 25, 75, 20, 90, 70, 100, 95

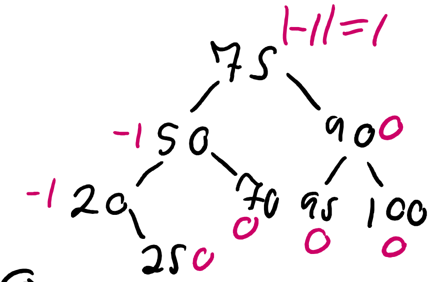
① Insert all the elements



② Determine the Balance Factor for all nodes

③ Determine the source sub tree for imbalance

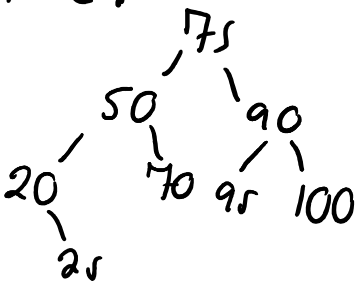
④ Rotate for balance
- Left Rotation of 50 & 75



⑤ Check balance again

⑥ Tree is balanced Final AVL tree

Final tree:



Question 4:

Please indicate the computational complex of the below list-based priority queue.

	insertItem	removeMin	minKey	minElement
Unsorted list implementation	O(1)	O(n)	O(n)	O(n)
sorted list implementation	O(n)	O(1)	O(1)	O(1)

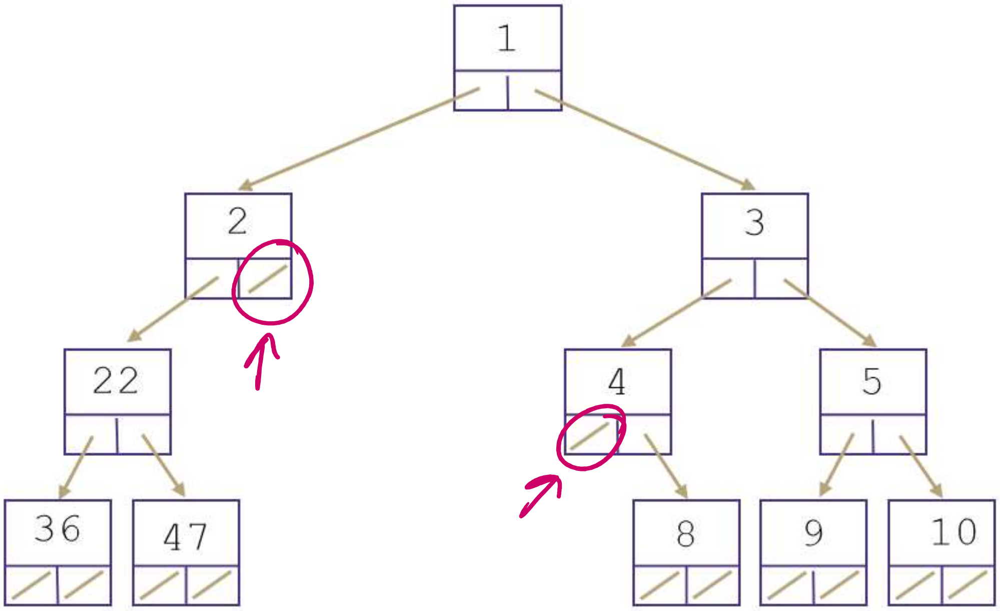
Question 5:

Given the below list-data, implement a selection sort for the priority queue.

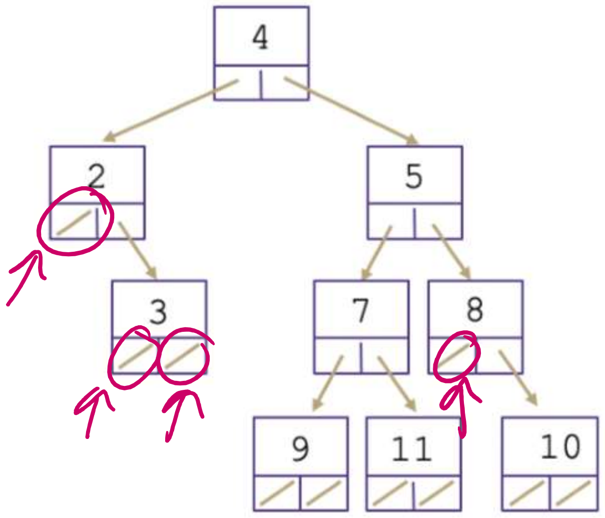
	Sequence S	Priority Queue P
Input	(7, 4, 8, 2, 5, 3, 9)	()
Phase 1: (a) (b) ... (g)	insert all elements ↓	(7) (7,4) (7,4,8,) (7,4,8,2) (7,4,8,2,5) (7,4,8,2,5,3)...
Phase 2: (a) (b) (c) (d) (e) (f) (g)	removeMin remove Min remove Min remove Min removeMin remove Min remove Min	(7,4,8, 5,3,9) : (2) (7,4,8,5, 9) : (2,3) (7, 8, 5, 9) : (2,3,4) (7,8,9) : (2,3,4,5) (8,9) : (2,3,4,5,7) (9) : (2,3,4,5,7,8) () : (2,3,4,5,7,8,9)

Question 6:

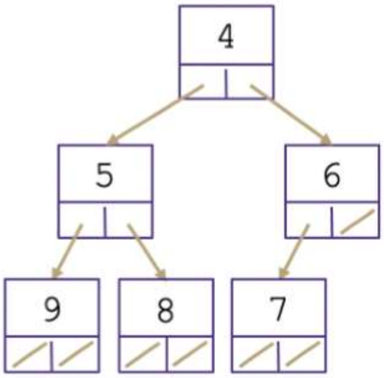
Is it a binary heap? Why or why not?



no, the binary heap must be a complete tree & this is an incomplete tree.



This is also not a binary heap since it also is an incomplete tree.

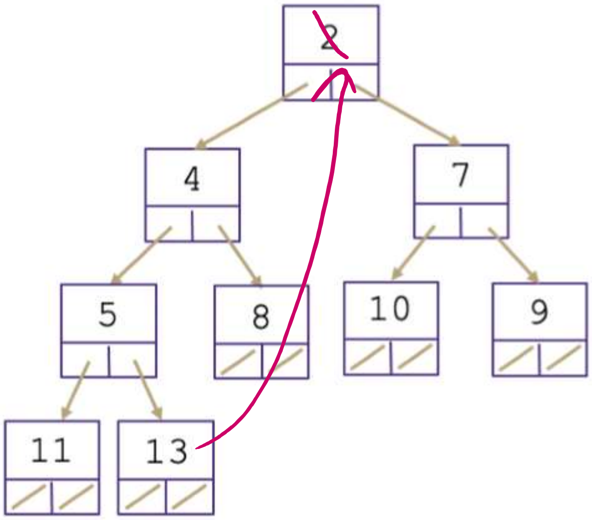


✓

this tree is valid, it meets all three binary heap requirements.

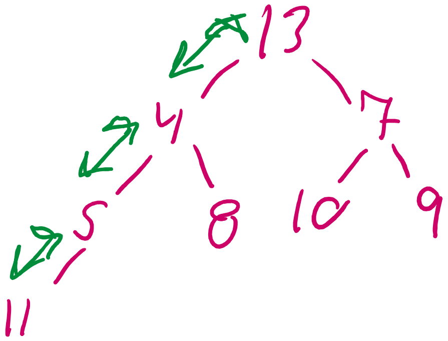
Question 7:

implement removeMin() for the below binary heap

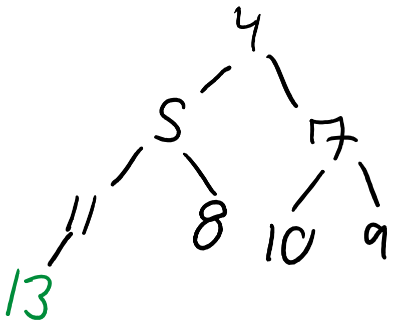


① removeRoot

② replace with last added

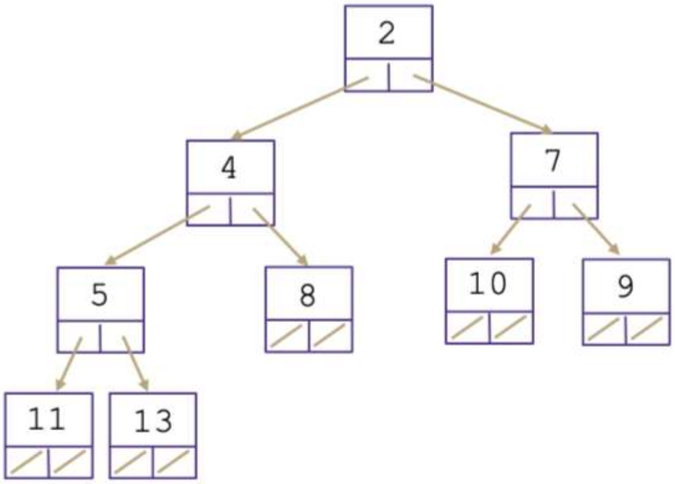


④ Percolate down

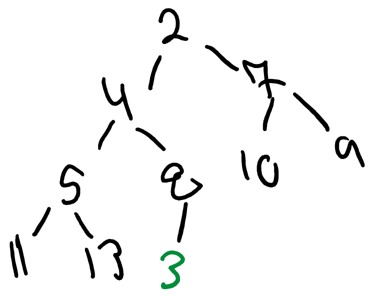


Question 8:

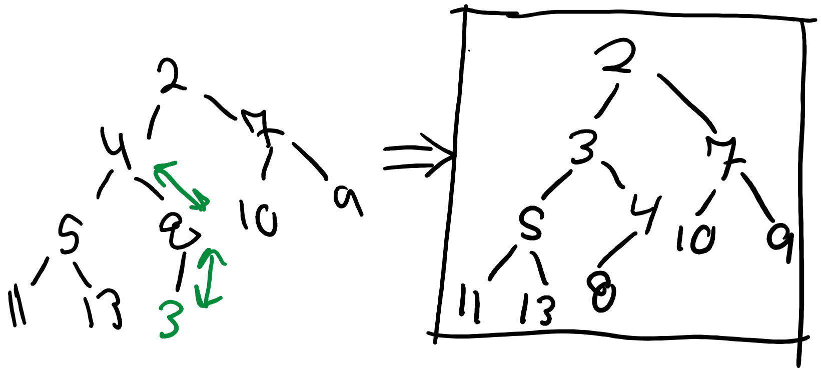
implement insert() for the below binary heap: insert a node with value 3.



① insert at end

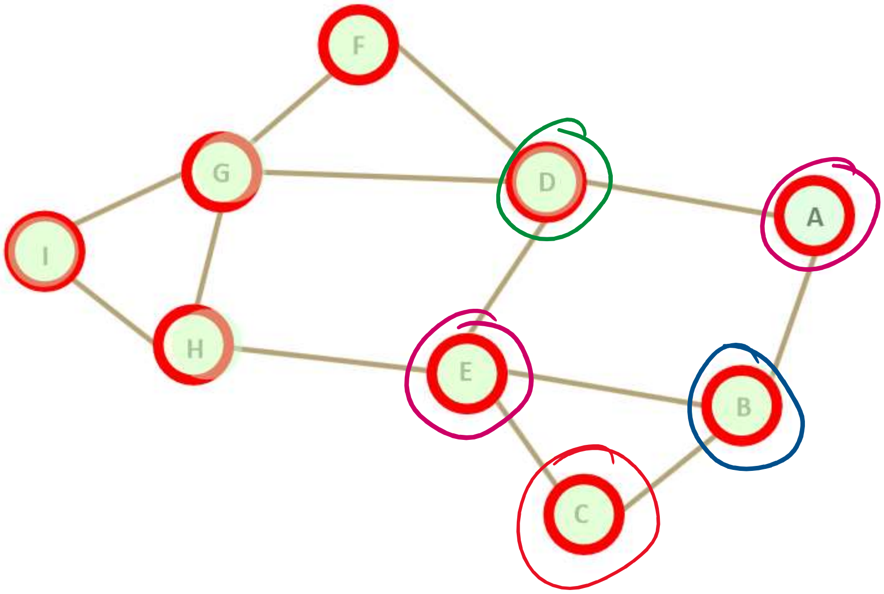


② Percolate up



Question 9:

fill out each step in the three parts: current node, queue, and visited, for implementing breadth first search for traversing a graph



① Current node: A
Queue: B, D
Visited: A

② current node: B
Queue: D, E, C
Visited: A, B

③ current node: D

Queue: E, C, F, G
Visited: A, B, D

④ Current node: E
Queue: C, F, G, H
Visited: A, B, D, E

5 Current node: C
Queue: F, G, H
Visited: A, B, D, E, C

6 Current node: F
Queue: G, H
Visited: A, B, D, E, C, F

7 Current node: G
Queue: H, I
Visited: A, B, D, E, C, F, G

8 Current node: H
Queue: I
Visited: A, B, D, E, C, F, G, H

9 Current node: I
Queue:
Visited: A, B, D, E, C, F, G, H, I

16