**CSE -4303 Data Structures** **Student id:**

**Quiz # 2** **Time: 20 min**

1. Assume you have queues with operations: enqueue(), dequeue(), isEmpty(). How would you use the queue methods to implement a stack, in particular, push() and pop() ? -5

| push() | pop() |
|---|---|
| Solution 1<br><br>push(e){<br>  q.enqueue(e)<br>}<br><br>pop(){<br>  tmpQ <-  new empty queue<br>  while  ! (q.isEmpty()){<br>  tmpE  <-  q.dequeue()<br>  if  ! (q.isEmpty())<br>  tmpQ.enqueue( tmpE )<br>  else{<br>  while  !( tmpQ.isEmpty())<br>  q.enqueue( tmpQ.dequeue())<br>  return tmpE<br>  }<br>  }<br>} | Solution 2<br><br>Here the idea is similar, but now push does most of the work.<br><br>pop(){<br>  q.dequeue()<br>}<br><br>push(e){<br>  make a new empty queue qTmp<br>  qTmp.enqueue(e)<br>  while !q.isEmpty(){<br>  qTmp.enqueue( q.dequeue() )<br>  }<br>  q <- qTmp<br>  return qTmp<br>} |

2. Fill the following table for different algorithms with appropriate Big-O or Big-theta notation. -4

| Algorithm | *Best Case* | *Average Case* | *Worst Case* |
|---|---|---|---|
| **Merge Sort** | $\Theta(n \ln(n))$ | $\Theta(n \ln(n))$ | $\Theta(n \ln(n))$ |
| **Quick Sort** | $\Theta(n \ln(n))$ | $\Theta(n \ln(n))$ | $O(n^2)$ |
| **Binary Search Tree, findMin** | O(1) | O(lg n) | O(n) |
| **Selection Sort** | $O(n^2)$ | $O(n^2)$ | $O(n^2)$ |

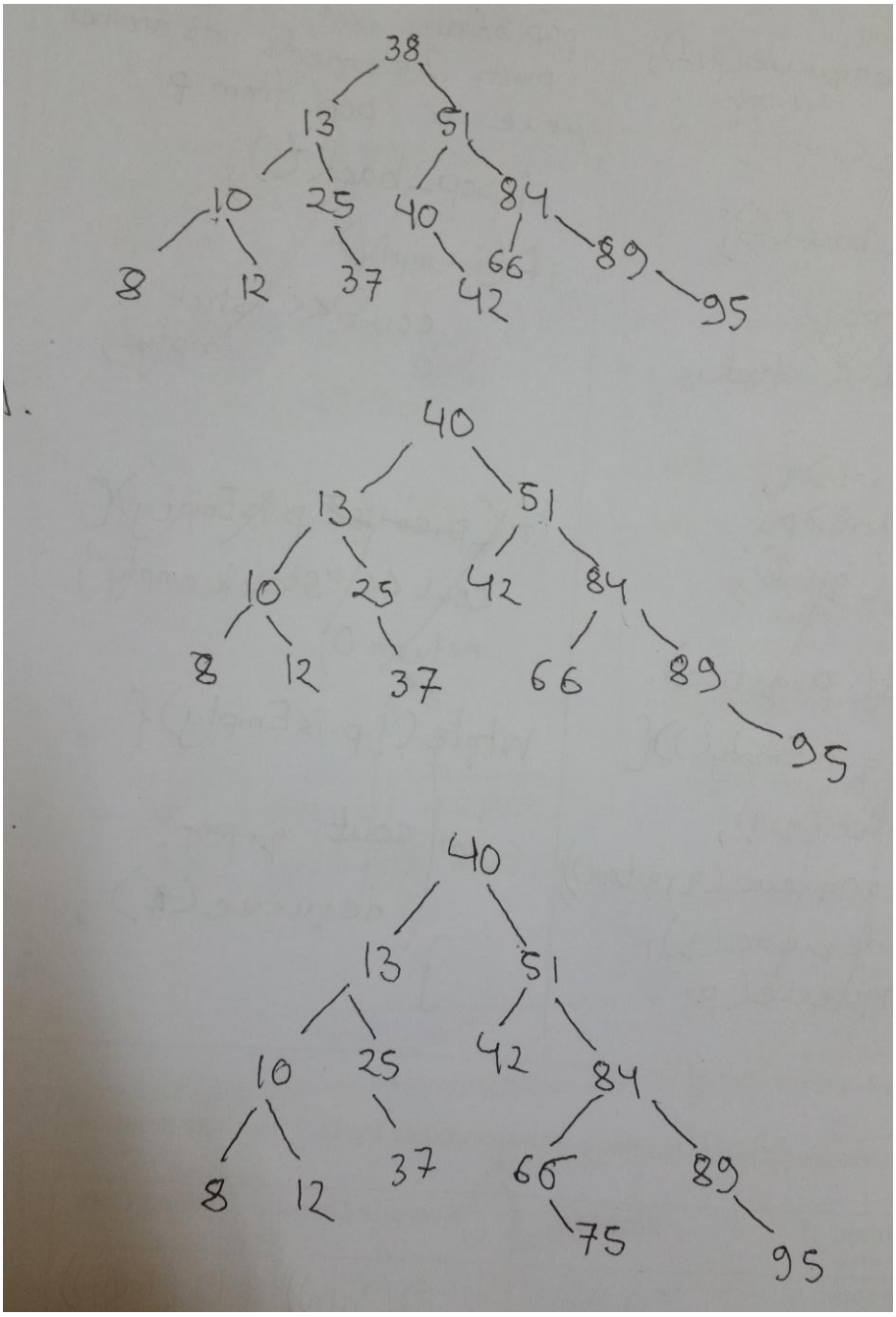
3. In the given order, insert these objects into an initially empty binary search tree:

**38 13 51 10 12 40 42 84 25 89 37 66 95 8**

Now perform the following operations in this binary search tree and draw the updated tree. -6

A. Erase 38

B. Insert 75




4. Write down your expected mark of Quiz 2 (out of 15). ☺☺☺ (**2 Bonus mark if it is matched**)