Boris Ayyildiz 1901042252 CSE 222 Midterm Examination B.Ayyildiz

I hereby pledge on my honor that I will strictly adhere to academic integrity codes and the work done on this examination is solely my own and I will not receive/give any help from/to anybody or source during this examination.

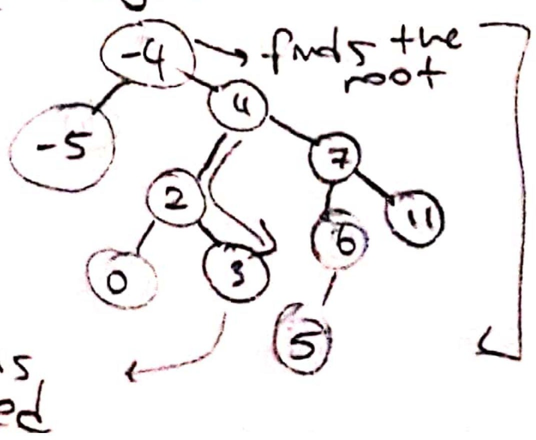
4) E is a generic type for questions 4 and 5

```
public E findPred(E value){
    if(root ==null) return null;
    if(root.compareTo(value) > 0 && root.left !=null)// value is lesser than the root
        return root.left.findPred(value);
    else if(root.compareTo(value) <0 && root.right !=null)//greater than the root
        return root.right.findPred(value);
    else{ // we found the root
      BinarySearchTree<E> iter= this.clone();
      if(iter.left ==null) return null //no child that is smaller
      iter=iter.left;
      while(true){
        if(iter.right ==null) return iter.root;
        iter =iter.right;
    }
```

Tbest = Θ(1) → if bst has only 1 element, it returns null

Tworst= O(logn) → number of searchs depends on the height of the tree

val = 4



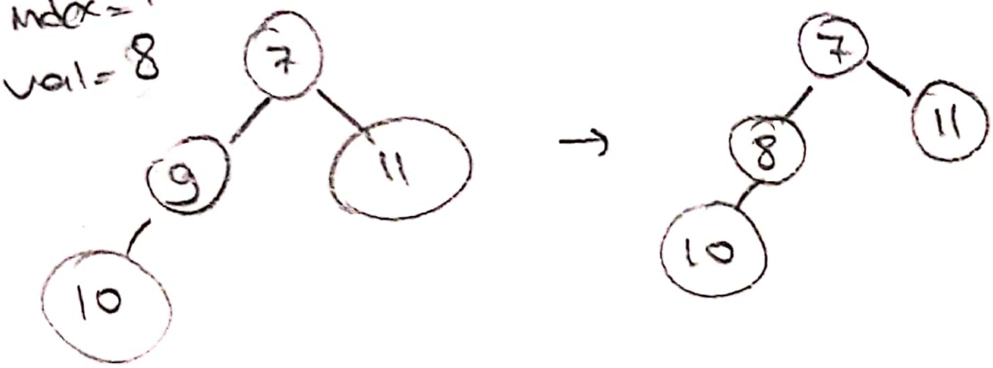
5) Let`s assume this is a min-heap

```
public boolean update(int index, E val){
   if(this.size() <= index) return false;
   arr[index] = val;
   int parent=(index-1) / 2;
   int left = 2*index+1;
   int right= 2*index+2;
   int child;
 while( parent >=0&& arr[index].compareTO (arr[parent]) <0){ less than
                                                             parent
     E temp = arr[parent];
     arr[parent] = arr[index];
     arr[index] = temp;
     index = parent;
     parent= (index-1) /2;
 }
 if(arr[left].compareTo(arr[right])>0) child= left;
 else child= right;
 while (arr[index].compareTo(arr[child]) >0 && child
                                              <this.size()
 { E temp=arr[child];
    arr[child]=arr[index];
    arr[index]=temp;
    index=child;
    left=2*index+1;
    right=2*index+2;
    if(arr[left].compareTo(arr[right] >0 )child=left;
    else child=right;
 } return true;}
```

Barış Ayyıldız 1901042252 CSE 222 Midterm Examination B. Ayyıldız

$T_{best} = \Theta(1)$ → when the new value is not greater than the childs nor lower than the parent. Or when the index is out of bounds.

index = 1
val = 8



$T_{worst} = O(\log n)$ Depends on the height of the tree