Problem Set 8, Part I

Problem 1: Checking for keys below a value

1-1) The time efficiency doesn't depend on the keys in the tree.
Based on the code, it has to trace through the whole tree which means it has to begin at root and go all the way to the bottom of the left of the tree. The right is the same as the left. Even if we find the target key, we still have to trace the whole tree. The notation of this code is always O(n).

1-2)

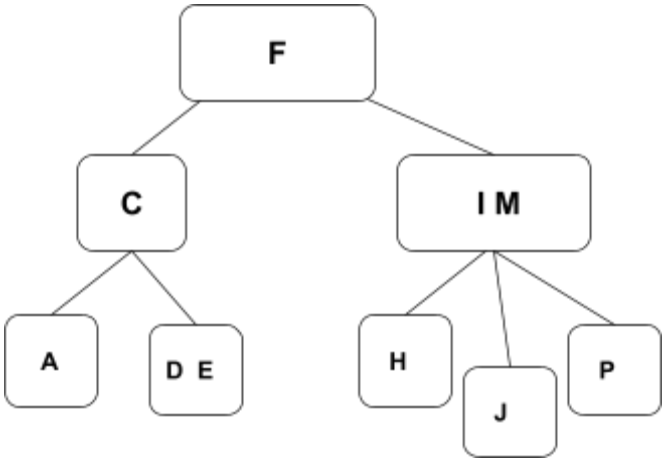
```
private static boolean anySmallerInTree(Node root, int v) {
	if(root == null){
		return false;
	}
	else{
		if(v<=root.key){
			return anySmallerInTree(root.left, v);
	}else{
		return true;
	}

}
```

1-3)
The time efficiency depends on the keys in the tree.The best case is the key of the root node is smaller than v, so the notation is O(1).

The notation of the worst case of the balanced tree is O(logn). This case happens when the node which is smaller than v is at the end of the tree and the height of two sides of the balanced tree is the same, so the notation is O(logn).

The notation of the worst case of the not balanced tree is O(n). This case happens when the node which is smaller than v is at the end of the tree, like one side is only one node which is the root, the other side is n nodes, so the notation is O(n).

Problem 2: Balanced search trees



Problem 3: Hash tables

3-1) linear

0	ant
1	flea
2	bat
3	cat
4	goat
5	dog
6	bird
7	bison

3-2) quadratic

0	
1	
2	
3	cat
4	goat
5	bird
6	bison
7	dog

3-3) double hashing

0	ant
1	bat
2	flea
3	cat
4	goat
5	bison
6	bird
7	dog

3-4) probe sequence: try 3%8 = 3; (3+5)%8=0; (3+2*5)%8=5; (3+3*5)%8=2 which is removed space; (3+4*5)%8=7; (3+5*5)%8=4 which is empty space - open, so eel will be inserted into the first removed position.

3-5) table after the insertion:

0	bobcat
1	
2	eel
3	fly
4	
5	koala
6	
7	penguin

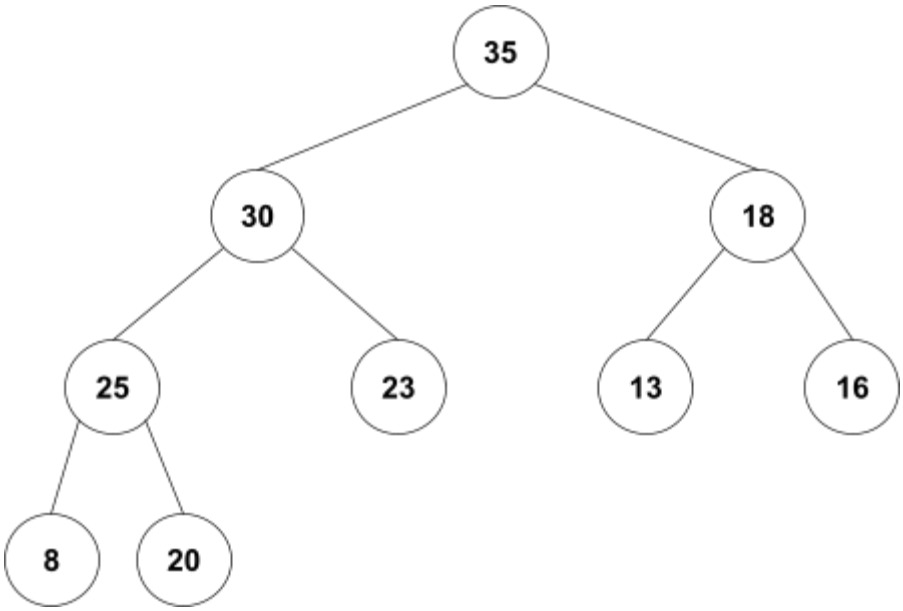
Problem 4: Complete trees and arrays

4-1) left child:23*2+1=47
right child:23*2+2=48
parent:(23-1)/2=11

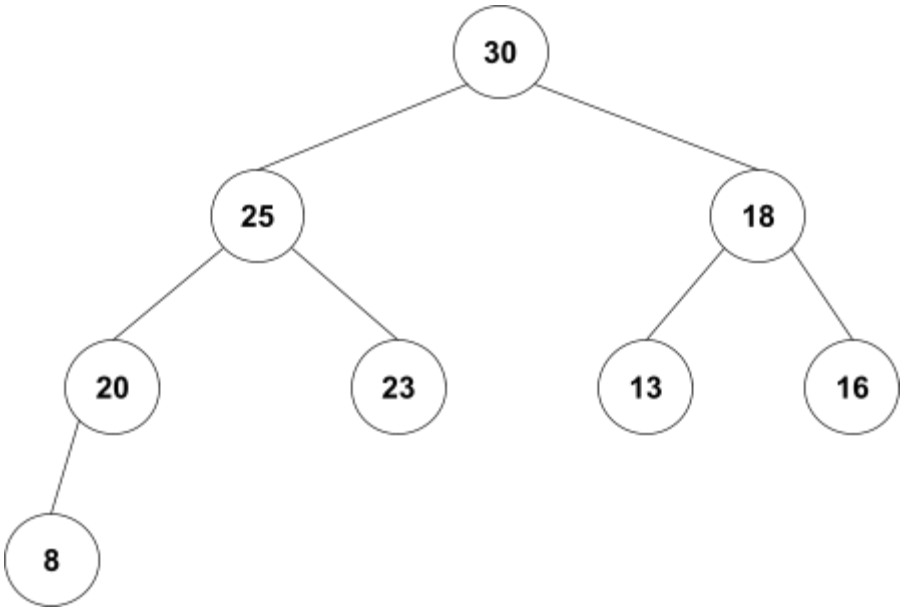
4-2) height = log2^(2023+1)-1=9.98299…=10

4-3) if the index of the leaf node is an odd number, it is the left child of its parent. if the index of the leaf node is an even number, it is the right child of its parent. Because the tree has 2023 nodes, the index of it is 2022 which is even, the rightmost leaf node in the bottom level is the right child of its parent.

Problem 5: Heaps
5-1)
after one removal



after a second removal
(copy your revised diagram from part 1 here, and edit it to show the result of the second removal)



5-2)