CS 323 Midterm 1 Practice

Name: **Net ID:** **Student ID:**

1 Complex code

What is the asymptotic time complexity of the following methods? Justify your answers.

1.1 Code 1 (5 points)

```
public static void smile(int X, int Y) {
    int k = X * X;
    while (k > 0) {
        k = k / 2;
        for (int h = Y; h < 3*Y; h += 2) {
            System.out.println(”:-)”);
        }
    }
}
```

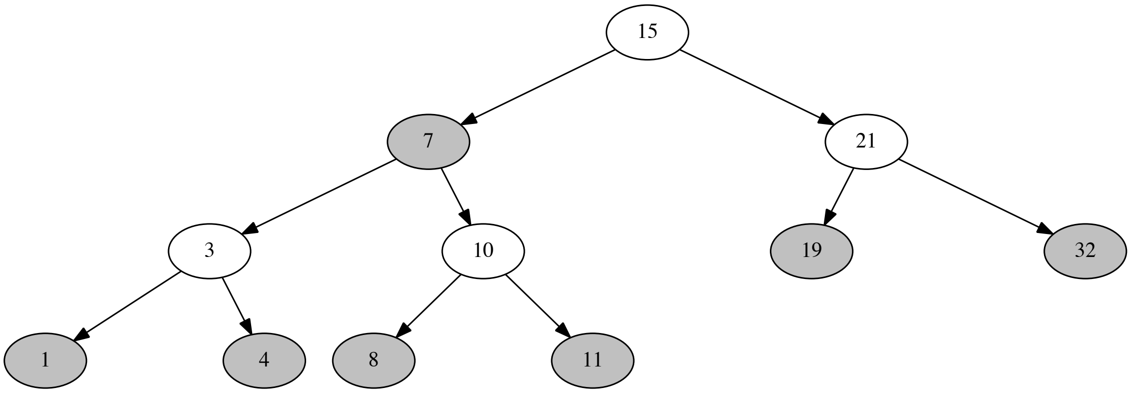
1.2 Code 2 (5 points)

```
public static void wink(int W) {
    System.out.println(”;-)”);
    if (W <= 0) {
        return;
    } else if (W % 2 == 1) {
        wink(W - 1);
        wink(W - 1);
    } else {
        wink(W / 2);
    }
}
```

2 Trees

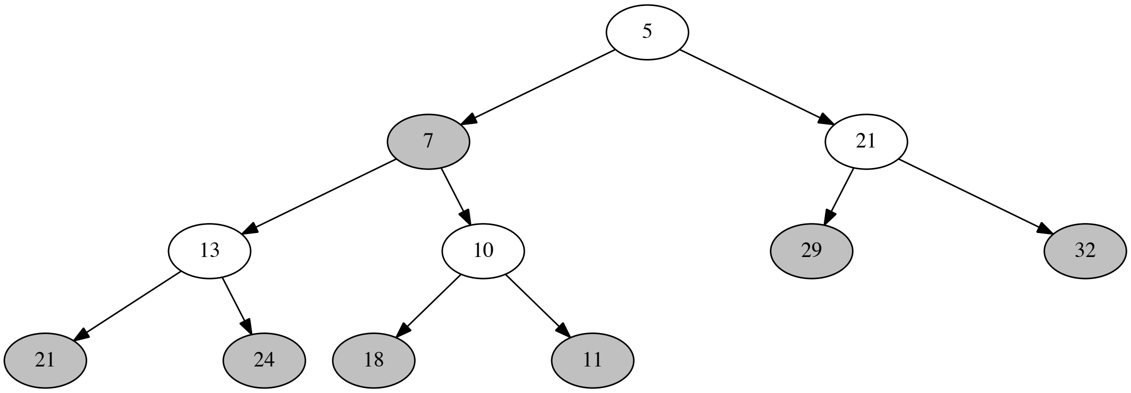
For each of the following trees, circle if it is: [A] a valid Binary Search Tree; [B] a valid Red-Black Tree; [C] a valid AVL Tree; [D] a valid Heap; [E] None of the above. Notice that you can select multiple options for each tree. In the figures, black nodes are white and red nodes are gray. NIL nodes are not drawn.

2.1 Tree 1 (3 points)



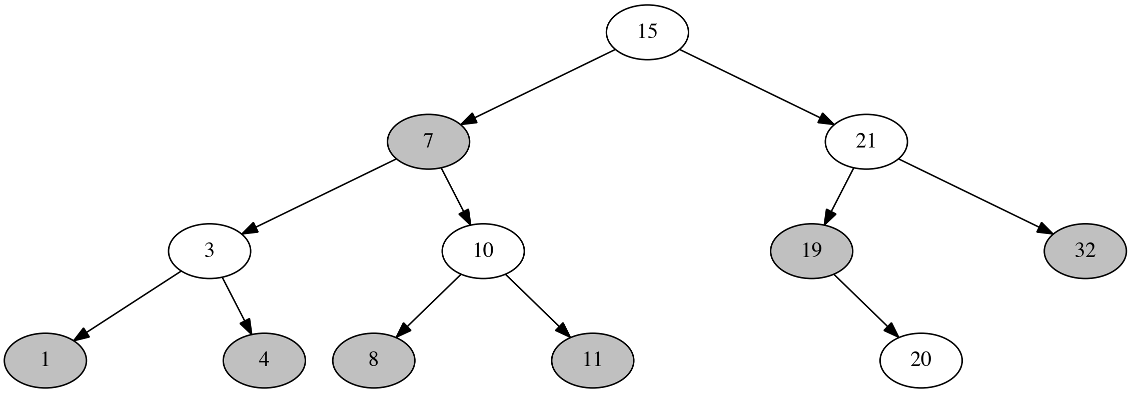
[A] [B] [C] [D] [E]

2.2 Tree 2 (3 points)



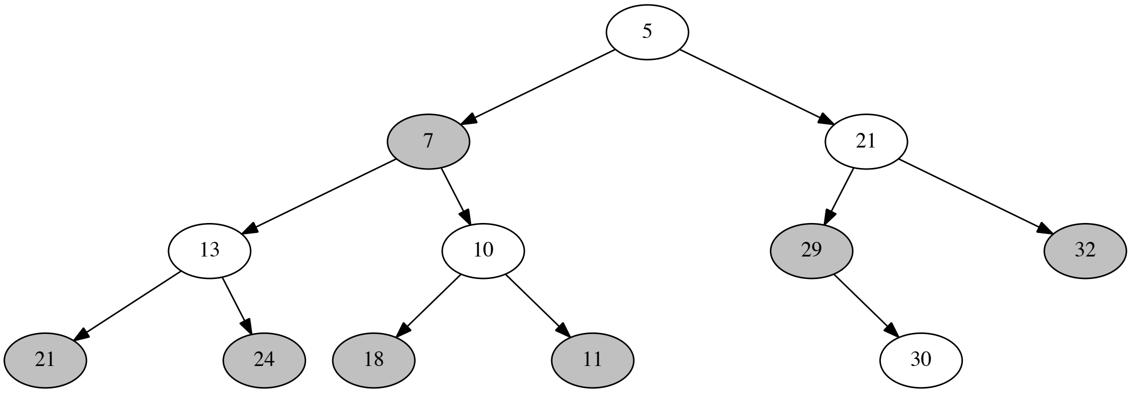
[A] [B] [C] [D] [E]

2.3 Tree 3 (3 points)



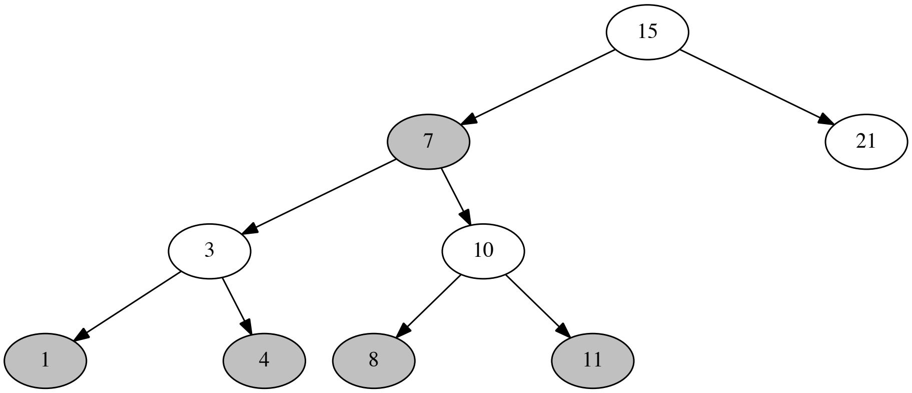
[A] [B] [C] [D] [E]

2.4 Tree 4 (3 points)



[A] [B] [C] [D] [E]

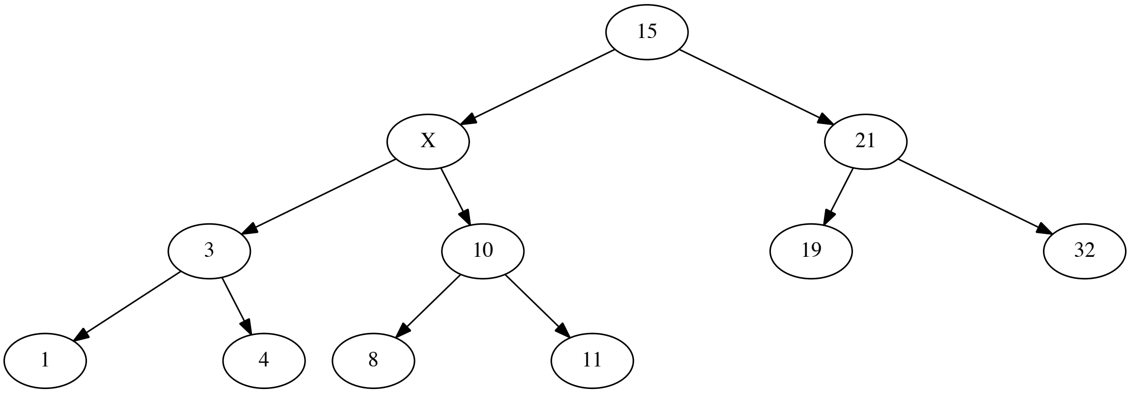
2.5 Tree 5 (3 points)



[A] [B] [C] [D] [E]

3 Tree rotations (10 points)

Perform a left rotation and a right rotation on node X of this tree, each time starting from the original tree. Draw the resulting trees.



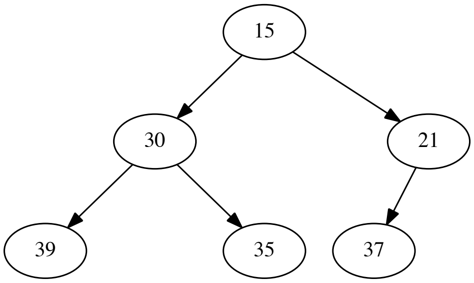
4 Disjoint-set forest (10 points)

You have a disjoint-set structure represented as a disjoint-set tree forest. Draw the complete state of the forest after each of the following operations (one drawing per line of code).

```
makeSet(A); makeSet(B); makeSet(C); makeSet(D); makeSet(E);
union(B, E);
union(A, C);
union(A, D);
union(C, B);
```

5 Heap tree to array (5 points)

Given the following tree representation of a heap, draw its corresponding array.



6 Heap array to tree (5 points)

Given the following array representation of a heap, draw its corresponding tree.

50	81	65	92	87	74

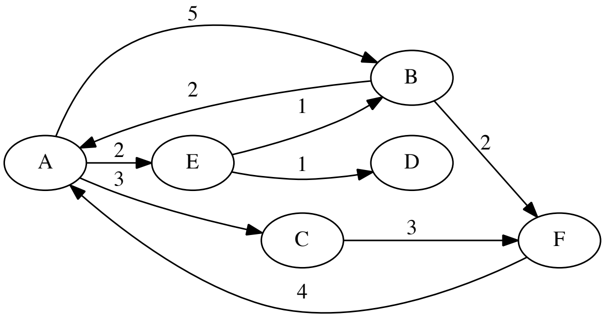
7 Build heap (10 points)

Using the Build Min Heap algorithm, transform the following array into a min heap. Show all the steps.

5	2	9	7	4	3

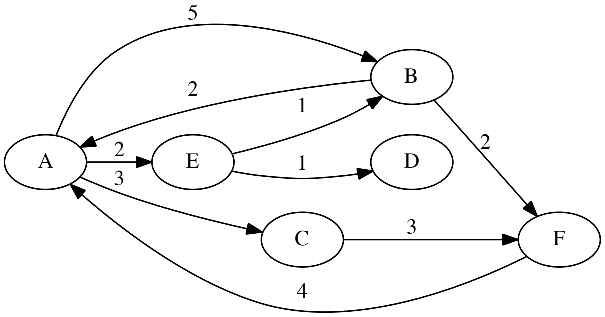
8 Graph representations (10 points)

Draw two representations of the following graph: (1) using adjacency lists, and (2) using an adjacency matrix.



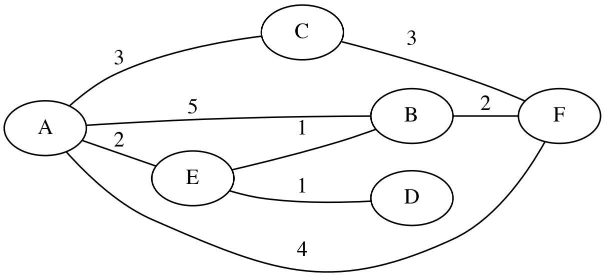
9 Graph BFS (5 points)

Show the order in which the vertices of the following graph are visited by the Breadth First Search algorithm starting from vertex A. Adjacent vertices are enqueued in alphabetical order.



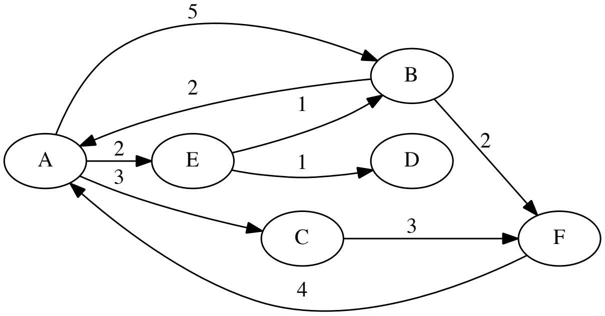
10 Kruskal (10 points)

Using Kruskal's algorithm, find the minimum spanning tree for this graph starting from vertex A. Show all the steps.



11 Bellman-Ford (10 points)

Using the Bellman-Ford algorithm, find the shortest paths from vertex A to all the other vertices of this graph . Show all the steps.


5
B
2
1
2
A
2
E
1
D
3
C
3
F
4