

COP 3503 Section 11 Final Exam Review Questions

1) Show the result of inserting the following into a red-black tree in order:

4, 5, 2, 1, 7, 9, 3, 6, 8

2) Behold the new, and probably doomed to fail, crypto currency! The Arupcoin. You've decided to invest in this coin before the bottom inevitably falls out. Due to your revolutionary prediction algorithm, you have an array of prices where $prices[i]$ is the price of Arupcoin on the i th day. On each day may decide to buy and/or sell Arupcoin. You may hold at most one Arupcoin at a time, you can buy or sell on each day. Given the function header, write a greedy solution that returns the maximum profit you can achieve. Note the solution shouldn't take any more than 25 lines with proper whitespace.

Example:

Input : {7,1,5,3,6,4}

Output: 7

Explantation: Buy on day 2, when price == 1 then sell on day 3, when price == 5, for a profit == $5 - 1 == 4$.

Then buy on day 4, when price == 3, then sell on day 5, when price == 6, for a profit == $(6 - 3) + 4 == 7$.

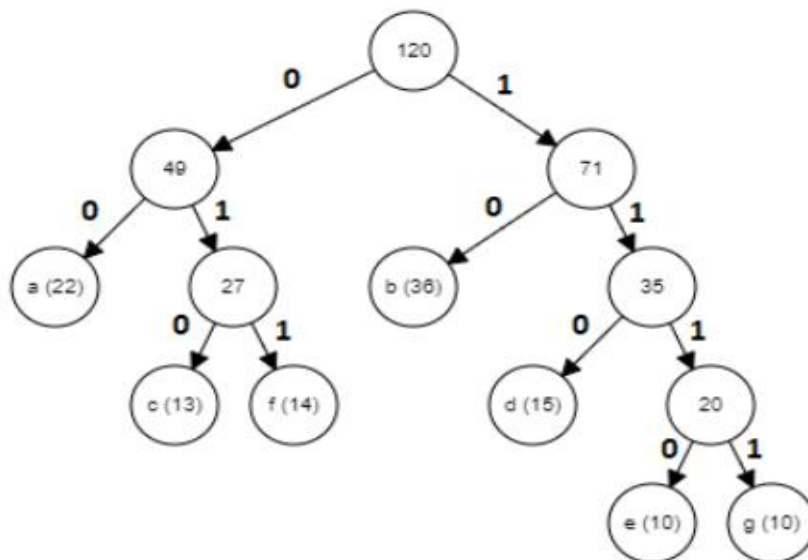
```
public static int maxProfit(int[] prices) {
```

```
}
```

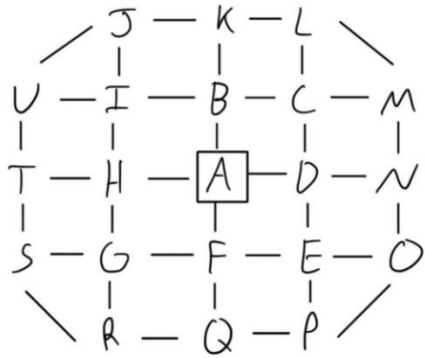
3) The following lines of code have some dependencies. Each line of code can be thought of as a vertex in a graph, and the edges denote the dependencies. If a topological sort is performed upon such a graph, then how many different sorts are there?

```
1. int a = stdin.nextInt();
2. int b = stdin.nextInt();
3. int c = stdin.nextInt();
4. int discriminant = b*b - ( 4*a*c );
5. double x1 = -1*b / (2*a) + sqrt(discriminant)/(2*a);
6. double x2 = -1*b / (2*a) - sqrt(discriminant)/(2*a);
7. System.out.printf("x = %0.2f or x = %0.2f, when %dx^2 + %dx + %d = 0", x1, x2, a, b, c);
```

4) Given the following Huffman Tree, determine the number of bits saved. The number next to a character represents its frequency. Each character was encoded using 3 bits.



5)

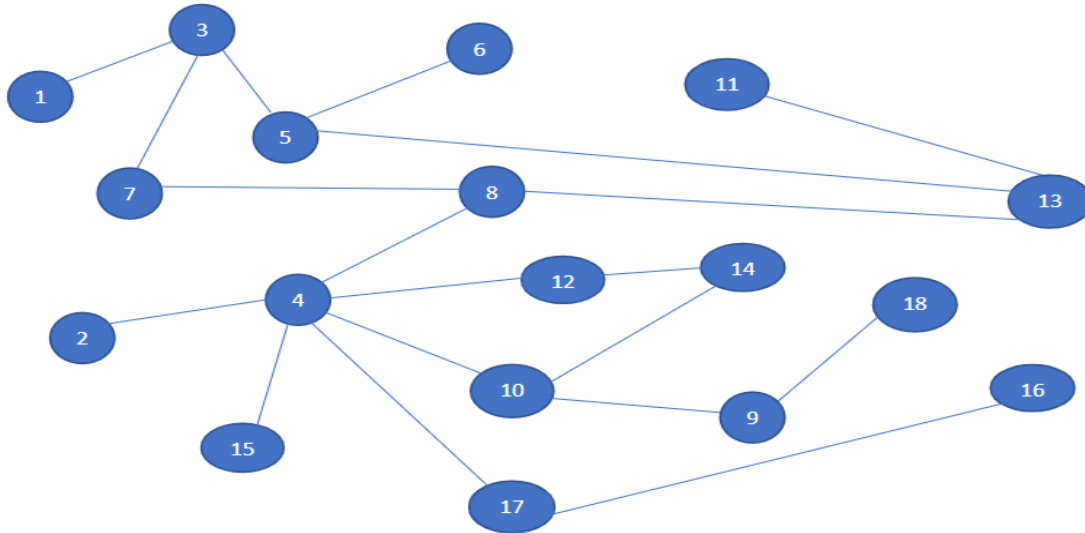


A. Show the order that all the nodes are visited when starting a Depth First Search (DFS) from vertex A. When iterating through the neighbor vertices of a given vertex, do so in alphabetical order.

B. What is the max depth of DFS starting from node A, using the rules in part A?

C. What is the last node visited on a DFS that starts from node K?

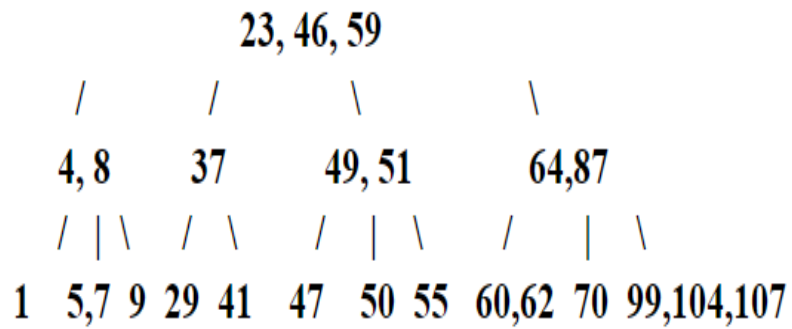
3) Perform a depth first search (DFS) and a breath first search (BFS) with the following constraints and write the order in which the vertices are visited. When given multiple vertices to traverse to, prioritize odd numbered vertices over even numbered vertices. When two or more vertices can be visited after the first constraint, prioritize the smaller vertex value over larger ones. The starting vertex for both searches is vertex 1.



Depth First Search

Breath First Search

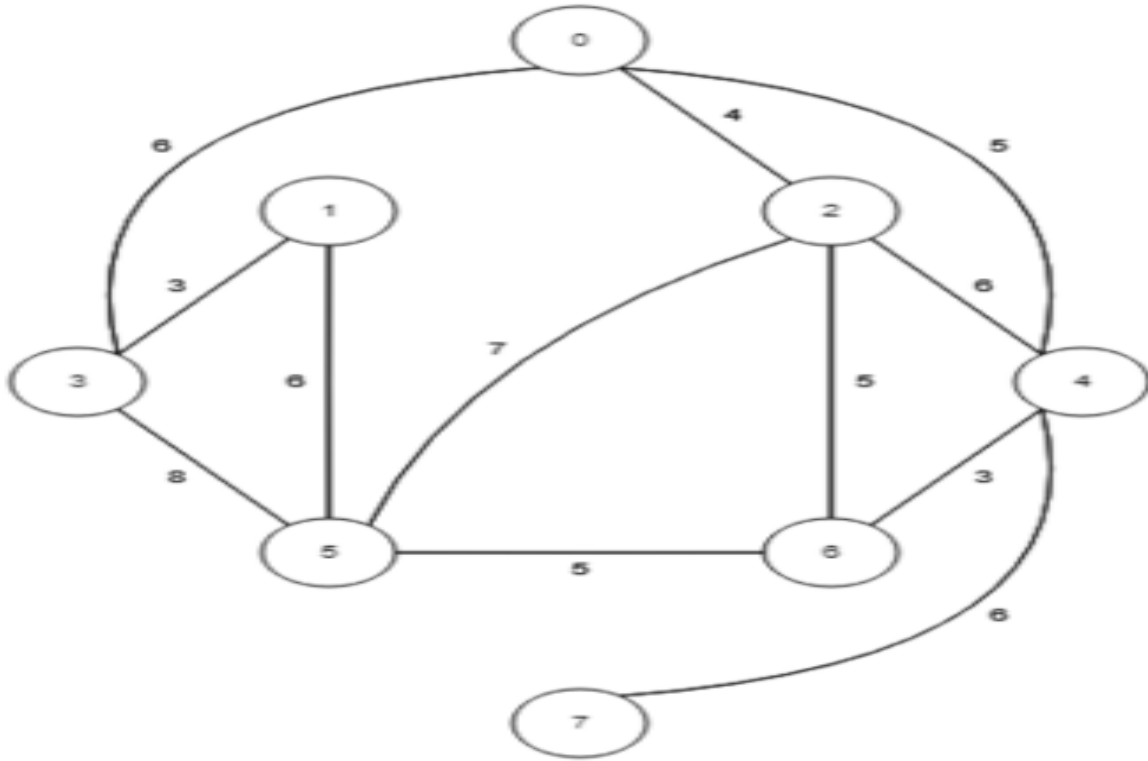
4) Given the following 2-4 Tree:



Perform the following actions and show the resulting tree after each action. Note that each instruction looks at the tree created by the previous instruction.

- 1) Insert 88
- 2) Delete 8
- 3) Insert 6

5) Using 0 as the start vertex for Prim's algorithm, find an MST of the following graph. Show the order in which the edges were considered to be added to the MST and whether or not they were actually added.



6) The following 0-1 Knapsack code works but runs too slow due to repeated computation. Apply memoization with an array called memo so it no longer does repeated computation.

You can assume that memo is a 2D array that has been filled with the value -1 . It is global so it is unnecessary to place it into the function's arguments.

```
private int knapsack(int[] weights, int[] values, int i, int capacity){
    if(i >= weights.length) return 0;

    if(capacity - weights[i] >= 0){
        return Math.max(values[i] + knapsack(weights, values, i + 1, capacity - weights[i]),
            knapsack(weights, values, i + 1, capacity));
    }

    return knapsack(weights, values, i + 1, capacity);
}
```

```
private int knapsack(int[] weights, int[] values, int i, int capacity){
```

```
}
```

COP 3503 Section 13 Final Exam Review Questions

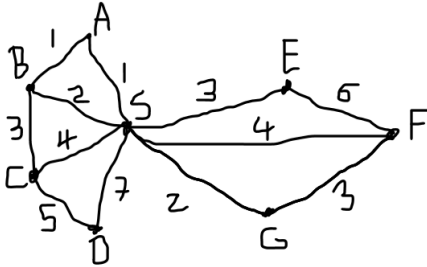
1) Show the state of a red-black tree after the following items have been inserted to an empty red-black tree, in this order: 20, 29, 8, 21, 22.

2) After eating a questionable Oreo that you found in a CS1 classroom, you discover that you can see into the future, so of course you decide to test it out by making a skip list. Using Java's random class, you foretell that the next 12 numbers between 0 and 1 are [.32, .73, .65, .93, .21, .53, .64, .26, .50, .41, .77, .48]. What will the skip list look like if you are storing the elements 3, 6, 5, 1, 23, 14 in order, and to ascend an element the random number must be greater than .50?

3) Which representation faster for inserting a new vertex into a graph, adjacency list or adjacency matrix? Why?

COP 3503 Section 14 Final Exam Review Questions

1) Draw the Minimum Spanning Tree for the following tree by using Prim's algorithm starting from node S.

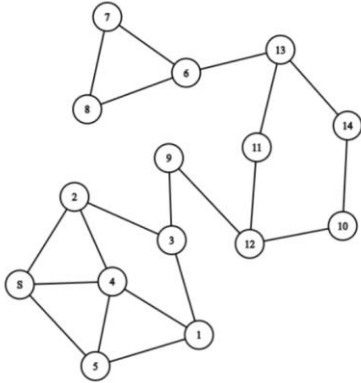


2) This question was omitted.

3) Perform a Radix sort on the following set of values:

123 321 781 555 902 319

4) Given the following graph, show the order the vertices get visited in a BFS starting at the vertex labeled 'S'. When enqueueing the neighboring vertices of a vertex, do so in numerical order.



5) Convert the following 2-4 tree into a red-black tree

