1 Minimum Spanning Trees

a) Perform Prim’s algorithm to find the minimum spanning tree of the following graph. Pick A as the initial node. Whenever there are more than one node with the same cost, process them in alphabetical order.

b) Use Kruskal’s algorithm to find a minimum spanning tree.

c) Bonus! There are quite a few MSTs here. How many can you find?
2 Dynamic Programming: Fibonacci

a) Write a recursive memoized version of the Fibonacci function. As a reminder, \( \text{fib}(n) = \text{fib}(n-1) + \text{fib}(n-2) \). \( \text{fib}(0) = 0 \) and \( \text{fib}(1) = 1 \). Hint: You may want to define a helper function

```java
public static int fib(int n) {
    // Implementation
}
```

b) What is the running time of your method?

3 Dynamic Programming: Maximum Subarray

You are given an array of integers, \( \text{A} \). Find the subarray with the maximum sum. Let’s suppose we were given an array containing the elements \( \{-2, 1, -3, 4, -1, 2, 1, -5, 4\} \). The maximum subarray is \( \{4, -1, 2, 1\} \) with a sum of 6. Note that the empty subarray is valid, with a sum of 0. For example, given \( \{-1, -2, -3\} \), you would return 0 for the subarray \( \{} \)

a) Sometimes, we can define a problem in terms of subproblems. What might be an appropriate subproblem for this problem? Hint: If we know the the maximum sum of the array ending at index \( i - 1 \), what do we know about the maximum sum of the array ending at index \( i \)?

b) Write an iterative method to solve the problem.

```java
public static int maxSubarraySum(int[] A) {
    // Implementation
}
```