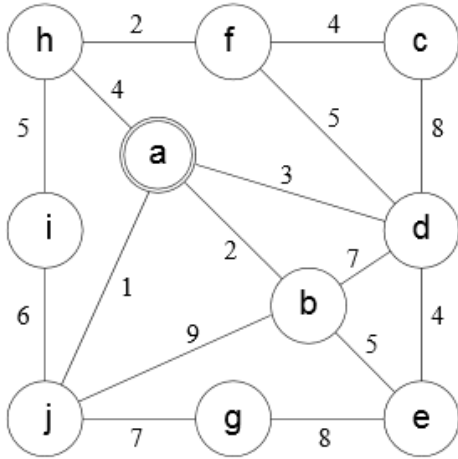


CS 325 - Homework Assignment 4 - Summer 2020

1. (3 points) Demonstrate Prim's algorithm on the graph below by showing the steps in subsequent graphs as shown in Figures 23.5 on page 635 of the text. What is the weight of the minimum spanning tree? Start at vertex a.



2. (6 points) Consider an undirected graph $G=(V,E)$ with nonnegative edge weights $w(u,v) \geq 0$. Suppose that you have computed a minimum spanning tree G , and that you have also computed shortest paths to all vertices from vertex $s \in V$. Now suppose each edge weight is increased by 1: the new weights $w'(u,v) = w(u,v) + 1$.

- (a) Does the minimum spanning tree change? Give an example it changes or prove it cannot change.
- (b) Do the shortest paths change? Give an example where they change or prove they cannot change.

3. (4 points) In the bottleneck-path problem, you are given a graph G with edge weights, two vertices s and t and a particular weight W ; your goal is to find a path from s to t in which every edge has at least weight W .

- (a) Describe an efficient algorithm to solve this problem.
- (b) What is the running time of your algorithm.

CS 325 - Homework Assignment 4 - Summer 2020

4. (5 points) Below is a list of courses and prerequisites for a factious CS degree.

Course	Prerequisite
CS 150	None
CS 151	CS 150
CS 221	CS 151
CS 222	CS 221
CS 325	CS 221
CS 351	CS 151
CS 370	CS 151
CS 375	CS 151, CS 222
CS 401	CS 375, CS 351, CS 325, CS 222
CS 425	CS 325, CS 222
MATH 200	None
MATH 201	MATH 200

- (a) Draw a directed acyclic graph (DAG) that represents the precedence among the courses.
- (b) Give a topological sort of the graph.
- (c) If you are allowed to take multiple courses at one time as long as there is no prerequisite conflict, find an order in which all the classes can be taken in the fewest number of terms.
- (d) Determine the length of the longest path in the DAG. How did you find it? What does this represent?

CS 325 - Homework Assignment 4 - Summer 2020

5. (12 points) Suppose there are two sides of a football "civil war": "Beavers" ("good guys") and "Ducks" ("bad guys"). Between any pair of teams, there may or may not be a rivalry. Suppose we have n players and we have a list of r pairs of rivalries.

(a) Give pseudocode for an efficient algorithm that determines whether it is possible to designate some of the team members as Beavers and the remainder as Ducks such that each rivalry is between a Beaver and a Duck. If it is possible to perform such a designation, your algorithm should produce it.

(b) What is the running time of your algorithm?

(c) **Implement:** Beavers vs. Ducks.

Input: Input is read in from a file specified in the command line at run time. The file contains the number of players, n , followed by their names, the number of rivalries r and rivalries listed in pairs.

Note: The file only contains one list of rivalries

Output: Results are outputted to the terminal.

- Yes, if possible followed by a list of the Beaver players and a list of the Ducks.
- No, if impossible.

Sample Input file:

```
5
Ace
Duke
Jax
Biggs
Stone
6
Ace Duke
Ace Biggs
Jax Duke
Stone Biggs
Stone Duke
Biggs Jax
```

Sample Output:

```
Yes
Beavers: Ace Jax Stone
Ducks: Biggs Duke
```

Submit a copy of your files including a README file that explains how to compile and run your code in a ZIP file to TEACH.