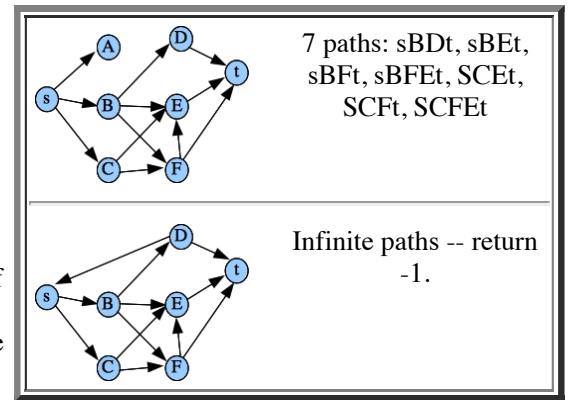


## Question 2: Problem Solving

You are given a directed graph with a starting node **s** and an ending node **t**. All nodes are reachable from **s**, and there are no edges from **t** to any other node in the graph. Your job is to tell me how many distinct paths there are from **s** to **t**. If there are an infinite number of paths, return negative one.

I want you to tell me the steps that you would take to solve this problem in terms of the algorithms that you have learned in this class. Be precise about how you are using each algorithm. Tell me the running time of each step. You may use **V** for the number of vertices, and **E** for the number of edges.

There are two examples to the right.



## Question 3: Know thy algorithms

For each of the algorithms to the right, give me a specification of a problem that can be solved with it. For example, if the algorithm is "Depth First Search", an answer could be: "Given an undirected graph, count the number of connected components in the graph. A connected component is a set of nodes where there is a path between every two nodes in the set." Note, the answer should not be "connected components." Yes, that's the name of a problem, but I want an actual specification of the problem to be solved. Your answer should with the word "Given," although if you can use a previous answer to define the problem, that's fine too.

- Dijkstra's algorithm
- Prim's algorithm
- Breadth-First Search
- Kruskal's algorithm

## Question 4: Dynamic programming

Explain the four steps of dynamic programming, using the fibonacci numbers as an example. In each step, I want to see the C++ code for **Fib**, whose class is defined to the right. You may add extra data to the class if necessary.

```
class Fib {
public:
    int find_fib(int n);
}
```

## Question 5: Network Flow

All of the questions that follow pertain to the graph to the right. Its source is **s** and its sink is **t**.

**Part A:** What is the maximum flow?

**Part B:** What is the minimum cut?

**Part C:** Suppose you are finding the maximum flow of the graph to the right using the greedy depth-first search to find augmenting paths. For each augmenting path, tell me:

- The path
- Its flow
- The resulting residual graph

**Part D:** Give me the augmenting paths if you use the Edmonds-Karp algorithm. Just tell me the paths -- I don't want to see the flow or residual graphs! I know that there are multiple ways to answer this. Just give me a legal answer.

I have worksheets for you. You may hand in a worksheet for your answer sheet if *you mark everything clearly and make it neat!* In other words, you may want to work out your answer first, and then write it on a clean sheet.

