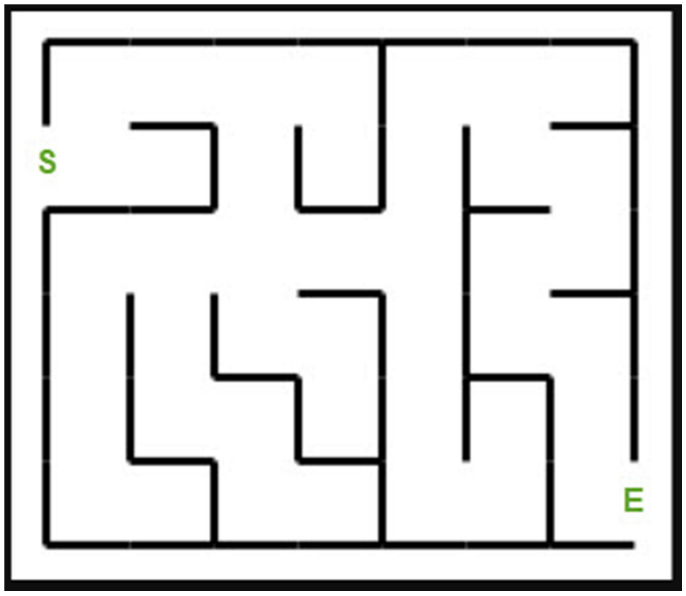
Week 12, Homework 2, Shahadat (19609)

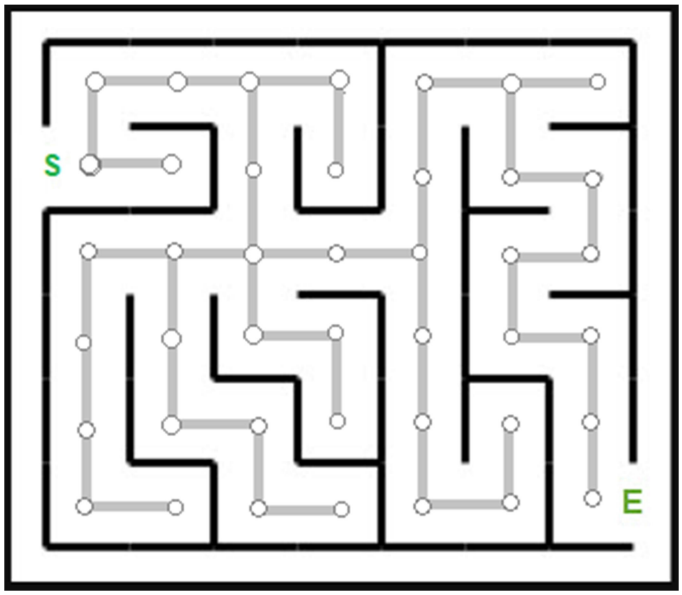
1. Project: Use Bellman Ford's Algorithm to find the shortest path of a maze.
    - Step 1: Similar to the previous question of finding the shortest path of the maze. But instead of using Dijkstra's Algorithm, you will use Bellman Ford's Algorithm.
    - Step 2: Comparing the performance of Dijkstra's Algorithm and Bellman Ford's Algorithm in solving this question by
        - Big-O comparison
        - Comparing how many steps are required to find a graph that has the shortest path.
            - Note:
                - A step is defined as either comparing two numbers or replacing a number.
                - You can count how many steps for Dijkstra's Algorithm on the created table.
                - Refer this example on counting the steps for Bellman Ford's Algorithm.
    - Step 3: Update your portfolio about the Maze project
    - Step 4: Submit the URL of your GitHub webpage as the homework answer.

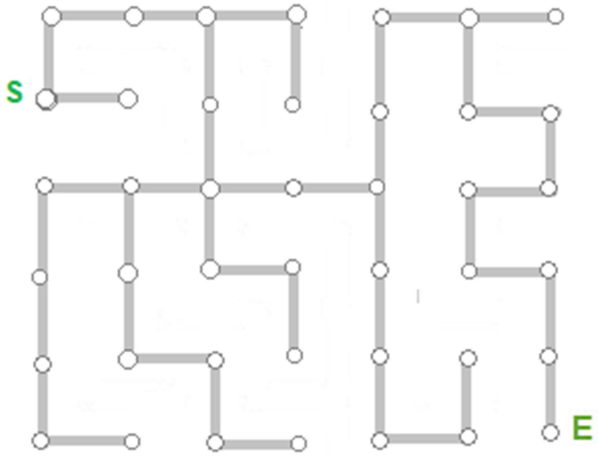
Answer:

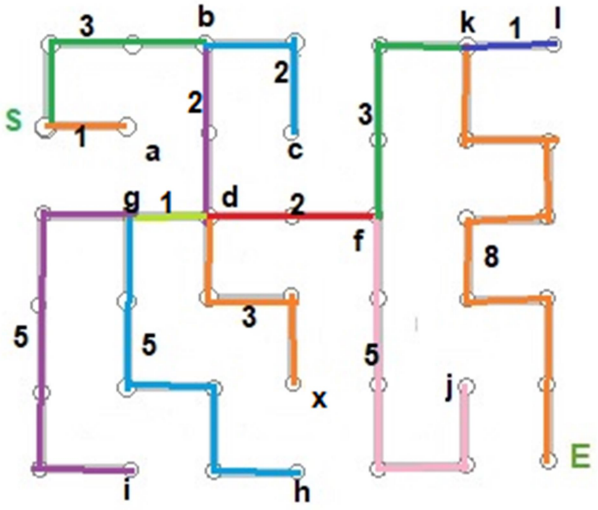
Find shortest path using Bellman Ford's Algorithm



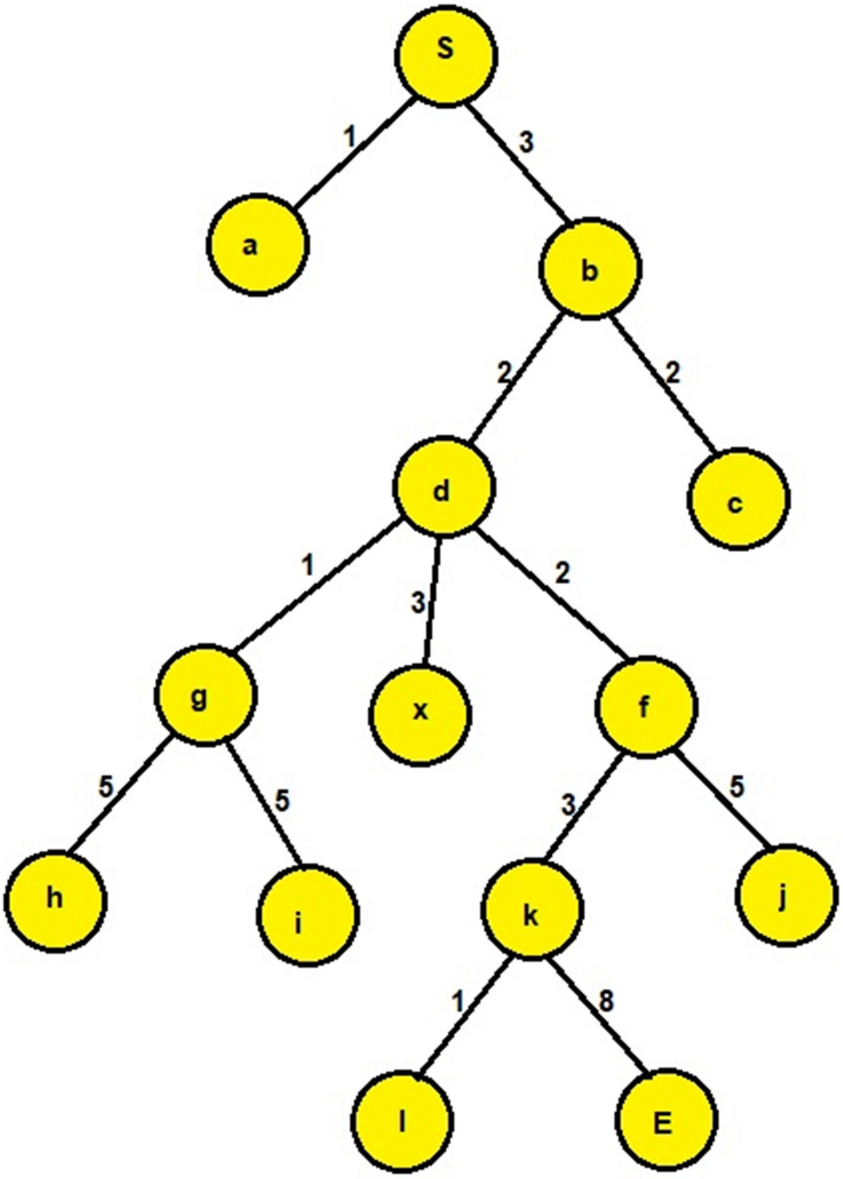
Step 1: Drawing target point on maze



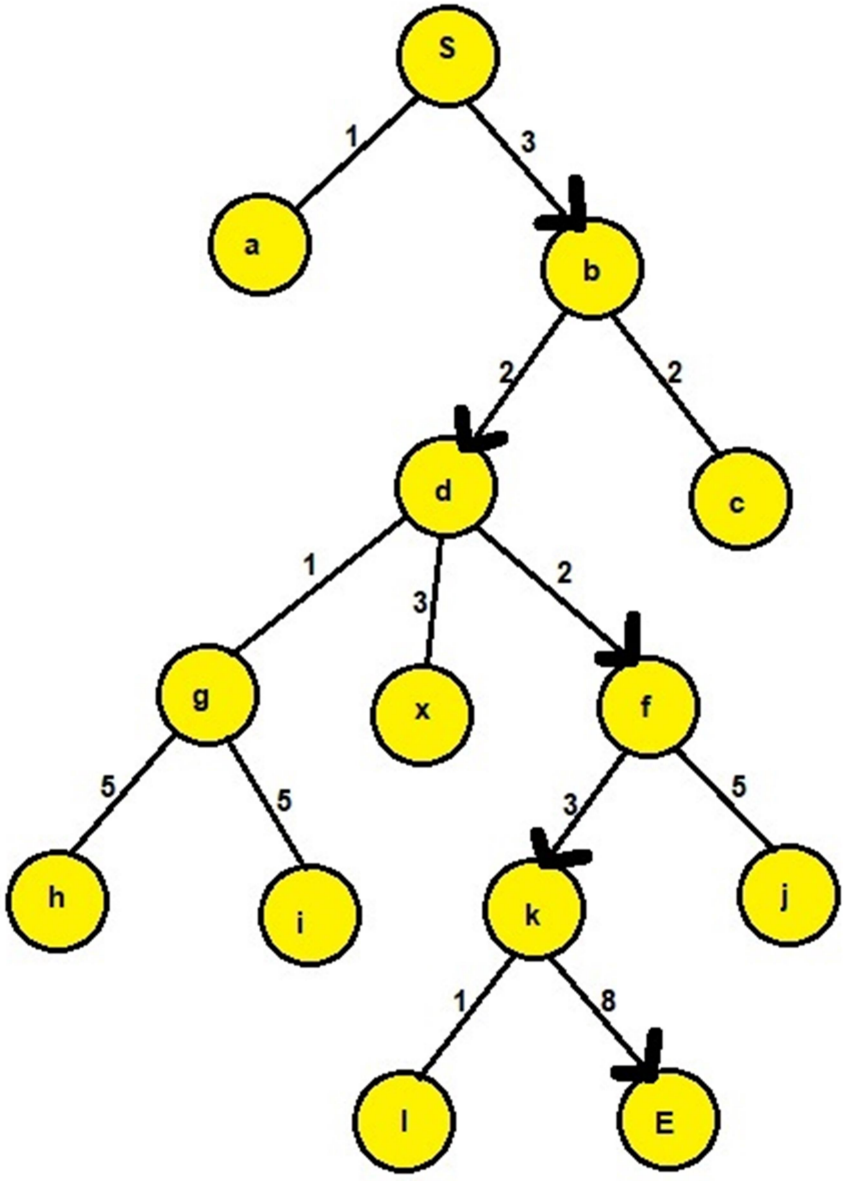




Step 2: Convert into tree



As this is tree structure, it has only one path



The path is: S -> b -> d-> f-> k-> E

The minimum distance of S to E is 18

Compare between Dijkstra's Algorithm and Bellman Ford Algorithm

Dijkstra's Algorithm	Bellman Ford Algorithm
Dijkstra's shortest path algorithm is O(ElogV) where: • V is the number of vertices • E is the total number of edges	Dijkstra's shortest path algorithm is O(VE) where: • V is the number of vertices • E is the total number of edges
Its need more time as per Big O analysis	Its need less time