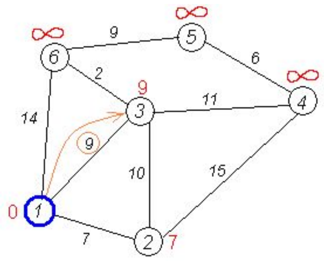
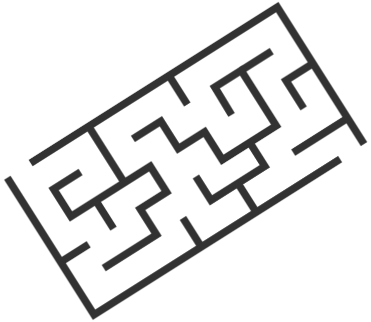


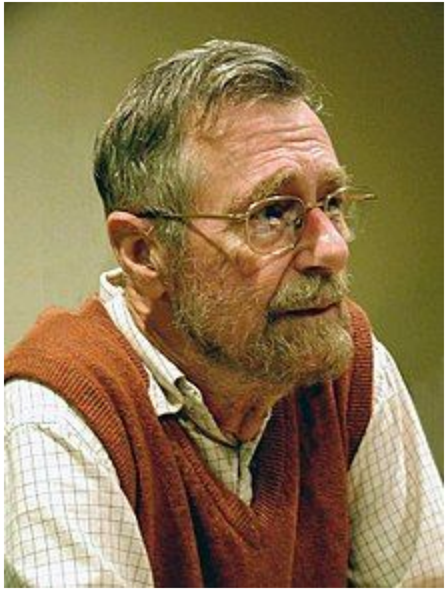
Dijkstra’s Algorithm to find the shortest path of the maze



Shoumya Singh

Table of Content

Edsger W. Dijkstra (1930-2002)



- Dutch Computer Scientist, programmer, software engineer, systems scientist, science essayist, and pioneer in computing science.
- Received Turing Award for contribution to developing programming languages.

Contributed to :

- Shortest path-algorithm, also known as Dijkstra's algorithm;
- Shunting yard algorithm; Banker's algorithm;
- Multithreaded programming ,multiprogramming system;
- Self-stabilization – an alternative way to ensure the reliability of the system etc.

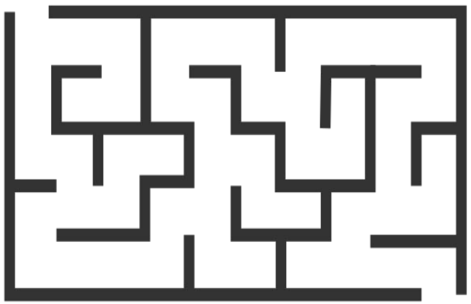
Introduction

- Dijkstra's algorithm solves the single-source shortest-paths problem on a directed weighted graph $G = (V, E)$, where all the edges are non-negative (i.e., $w(u, v) \geq 0$ for each edge $(u, v) \in E$).

- In simple words, algorithm for finding the shortest path from a starting node to a target node in a weighted graph is Dijkstra's algorithm.

- The algorithm creates a tree of shortest paths from the starting vertex, the source, to all other points in the graph.

What is a Maze?



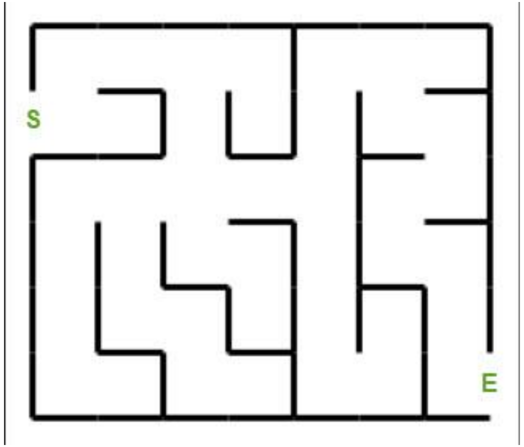
- A **maze** is a path or collection of paths, typically from an entrance to a goal.

- A maze can be viewed as a graph, if we consider each intersection in the maze to be a vertex, and we add edges to the graph between adjacent junctures that are not blocked by a wall.

- Our mazes will have random weights specified between any two adjacent intersection. These values can be thought of as the "cost" of traveling from one intersection to an adjacent one. The weights will be used for running Dijkstra's algorithm.

Table of Content

Design (Problem Statement)



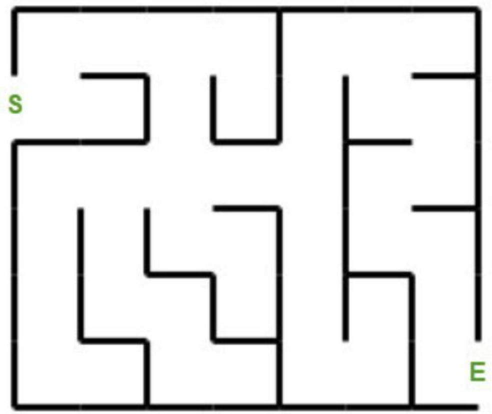
- Use Dijkstra's Algorithm to find the shortest path of the following maze .
- Convert the maze into a weighted Graph.
- Each node of the tree representation of the maze should be labeled sequentially and each edge should have a number indicating the distance.
- Apply the Dijkstra's Algorithm to find the shortest path in the weighted graph.

Table of Content

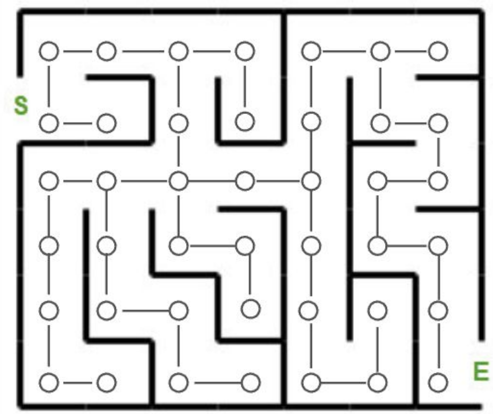
Implementation (Convert Maze to Weighted Graph)

- A maze can be viewed as a graph, if we consider each juncture (intersection) in the maze to be a vertex, and we add edges to the graph between adjacent junctures that are not blocked by a wall.

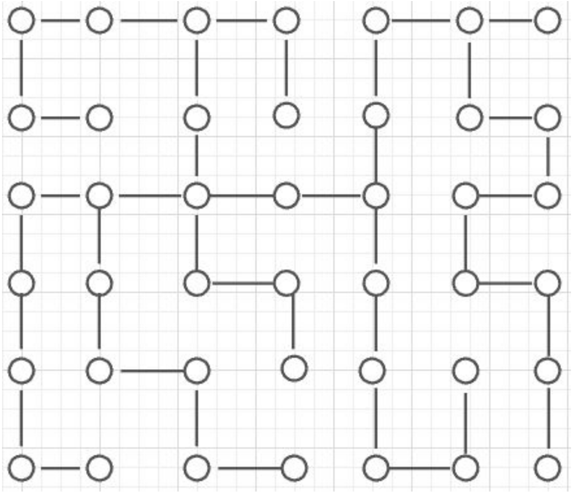
Step 1



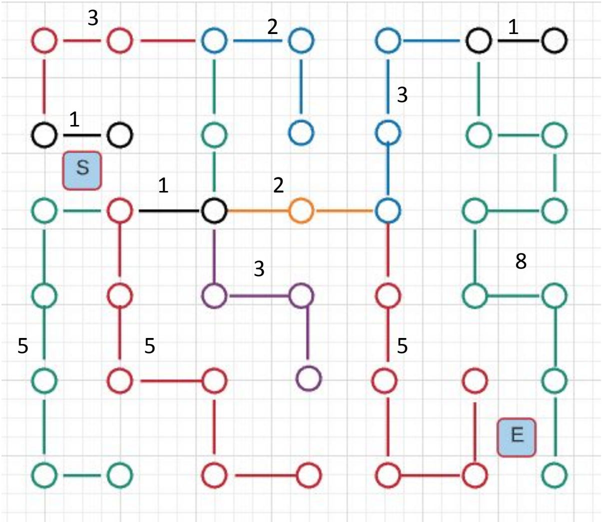
Step 2



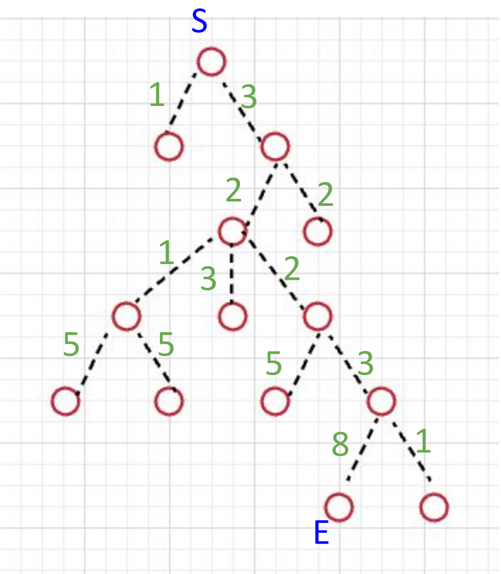
Step 3

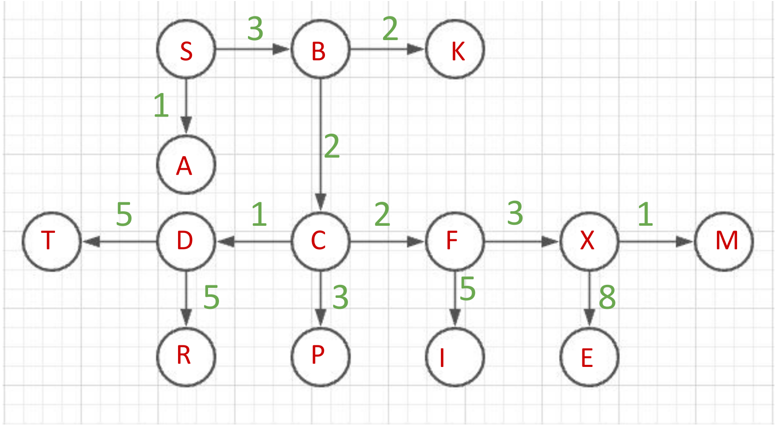


Step 4



Step 5





Apply Dijkstra Algorithm to Weighted Graph

Vertex(accumulated path)	Initial	Step1 S	Step 2 (S,B)	Step 3 (S,B,C)	Step 4 (S,B,C,D)	Step 5 (S,B,C,D, F)	Step 6 (S,B,C,D, F,X)	Step 7 (S,B,C,D, F,X,E)
	Next Step S	Next Step B	Next Step C	Next Step D	Next Step F	Next Step X	Next Step E	
S	0	0	0	0	0	0	0	0
B	In	3	3	3	3	3	4	4
A	In	1	1	1	1	1	1	1
C	In	In	5	5	5	5	5	5
K	In	In	5	5	5	5	5	5
P	In	In	In	8	8	8	8	8
D	In	in	In	6	6	6	6	6
F	In	In	In	7	7	7	7	7
R	In	In	In	In	11	11	11	11
T	In	In	In	In	11	11	11	11
I	In	In	In	In	In	12	12	12
X	In	In	In	In	In	10	10	10
M	In	In	In	In	In	In	11	11
E	**In**	**In**	**In**	**In**	**In**	**In**	**18**	**18**

V: the current visiting node
V: the next node to visit
V: this node has been visited

- Initially, all the vertices except the start vertex ***S*** are marked by ∞ and the start vertex ***S*** is marked by ***0***.

S is selected as the starting point for Step 1.

- From S, one can go to S or A or B
  - The accumulated cost on S is not changed. It is still 0.
  - The accumulated cost on A is 1.
  - The accumulated cost on B is 3.
  - 1 is smaller than 3.
    - But, B is selected as the starting point for Step 2 because A do have any further intersection.

- Stop if the destination node E is reached you will find the minimum distance of **E** from **S** is **18**.

The Shortest path is **S>B>C>D>F>X>E**

Table of Content

Enhancement Ideas

- The **enhancement Dijkstra's algorithm** is represented when we have more than two edges with the same weight such as (V→U) and (V→W)

- In this case we choose the vertex which have the maximum transition if (U) has the maximum transition than (W) we use the vertex (U) otherwise we choose (W).

Table of Content

Conclusion & Applications

- While it finds the shortest path with lower running time , It does not work with negative weight of edges in some networks. In this case, Bellman-Ford algorithm can be used which is very similar to Dijkstra's algorithm.

- Dijkstra’s Algorithm has several real-world use cases, some of which are as follows:
  - Digital Mapping Services in Google Maps
  - Social Networking Applications
  - IP routing to find Open shortest Path First
  - Robotic Path
  - Designate file server
  - Telephone Networks

Bibliography / References

1. https://en.wikipedia.org/wiki/Edsger_W._Dijkstra
2. https://brilliant.org/wiki/dijkstras-short-path-finder/
3. https://npu85.npu.edu/~henry/npu/classes/algorithm/tutorialpoints_daa/slide/shortest_paths.html
4. https://www.lucidchart.com/pages/
5. http://www.cs.umd.edu/class/spring2019/cmsc132-020X-040X/Project8/proj8.html
6. https://en.wikipedia.org/wiki/Maze#Solving_mazes
7. https://thesai.org/Downloads/Volume6No6/Paper_26-Using_Moore_Dijkstra_Algorithm_with_Multi_Agent_System.pdf
8. https://www.geeksforgeeks.org/applications-of-dijkstras-shortest-path-algorithm/