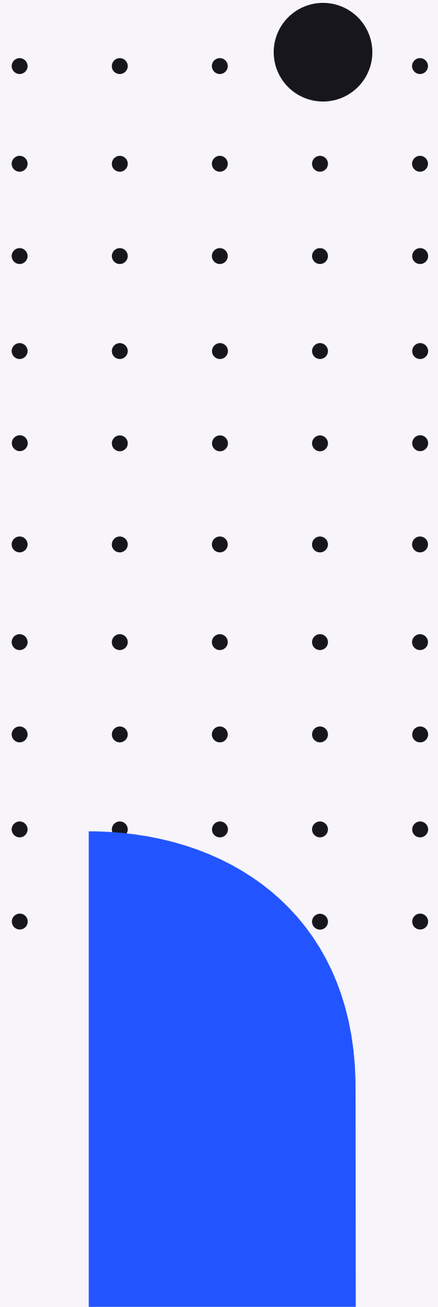
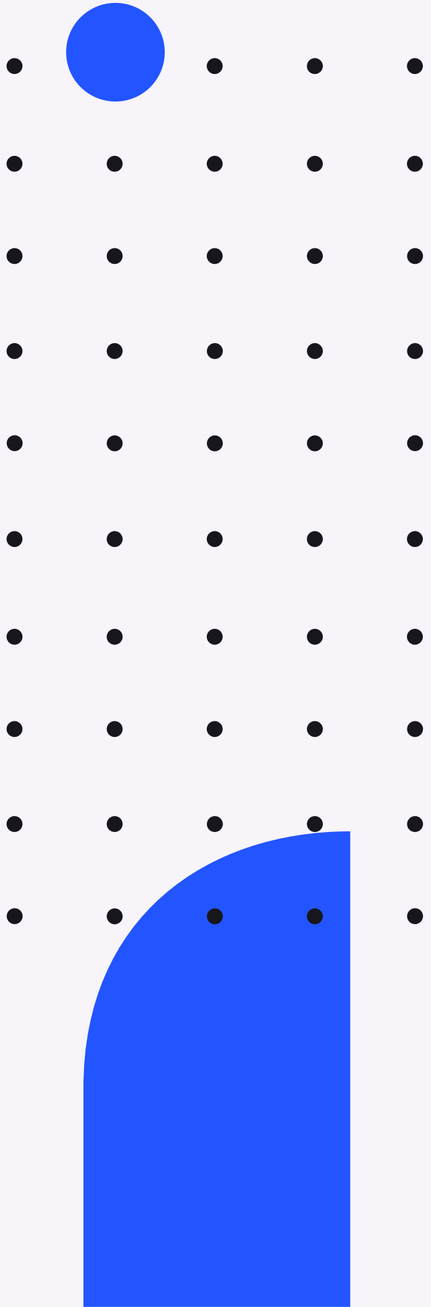
Depth-First Search (DFS)

Midterm project

Beisenbayeva Saule
Tagayeva Alina
Bisenbayeva Raushan
Abdrakhmanova Madina
Sagatova Ayaulym

Content



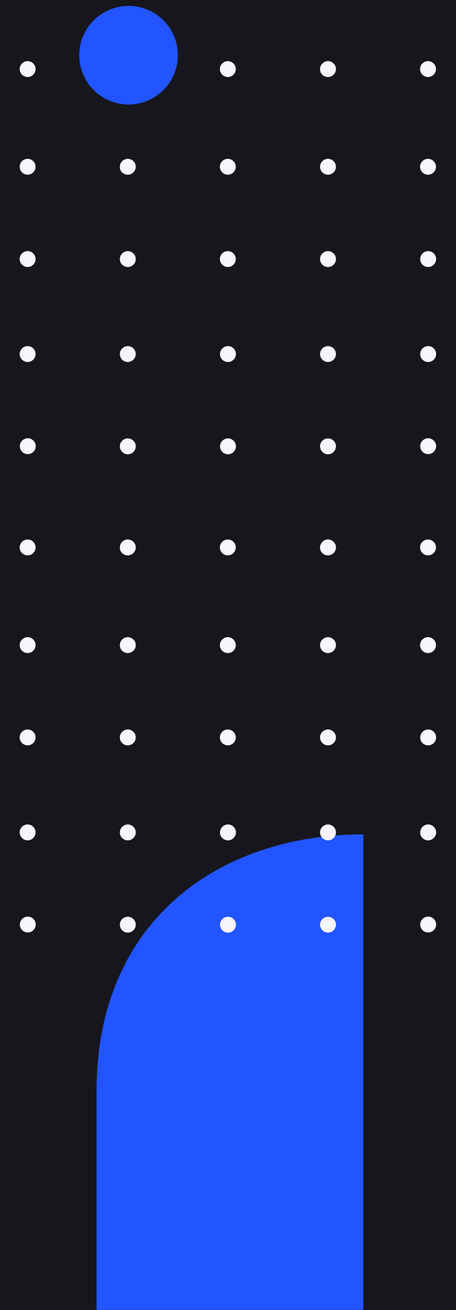


“go deep, head first”

-Ideology of DFS

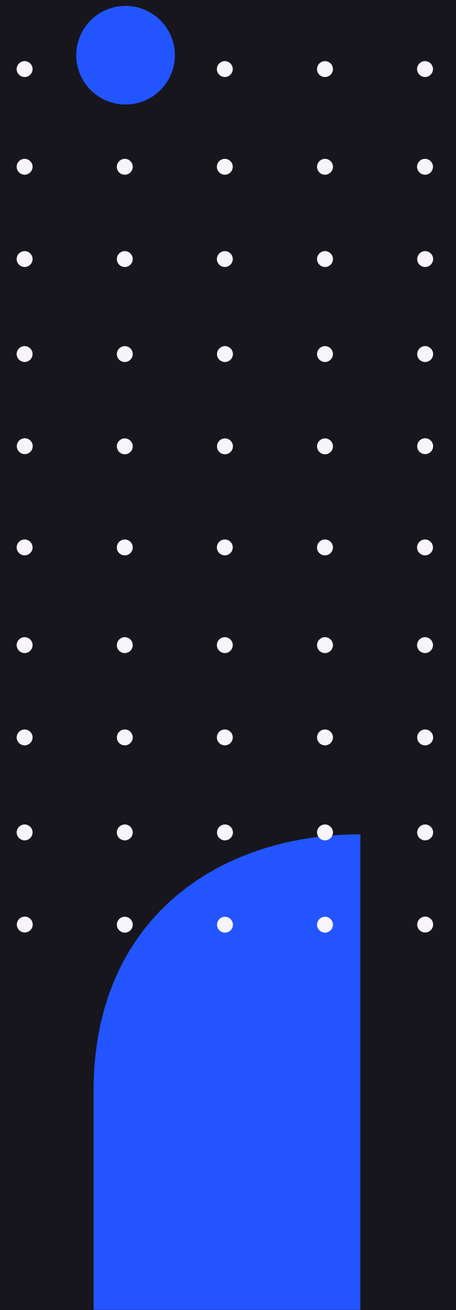
What is DFS?

- First graph algorithm
- Is an algorithm for traversing or searching tree or graph data structures
- One of the most fundamental algorithms in graph theory



What is graph traversal?

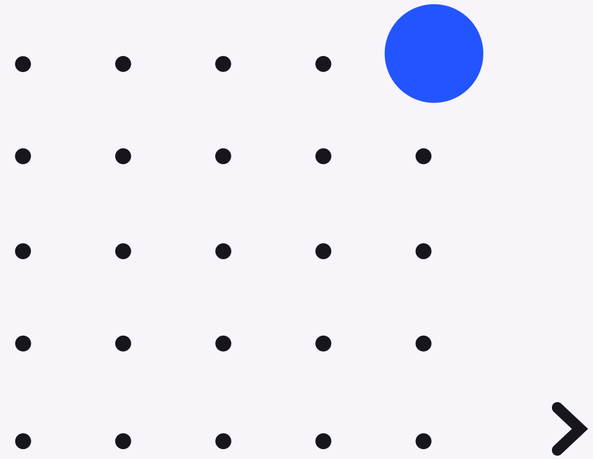
Graph traversal (also known as graph search) refers to the process of visiting (checking and/or updating) each vertex in a graph. Such traversals are classified by the order in which the vertices are visited. Tree traversal is a special case of graph traversal.



Time complexity

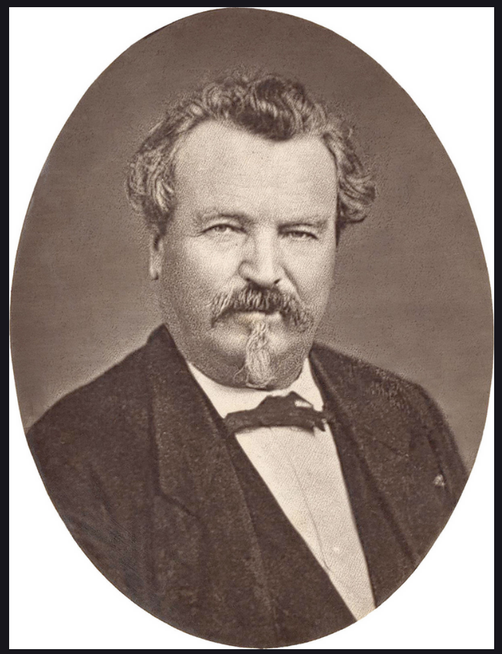
O(V+E)

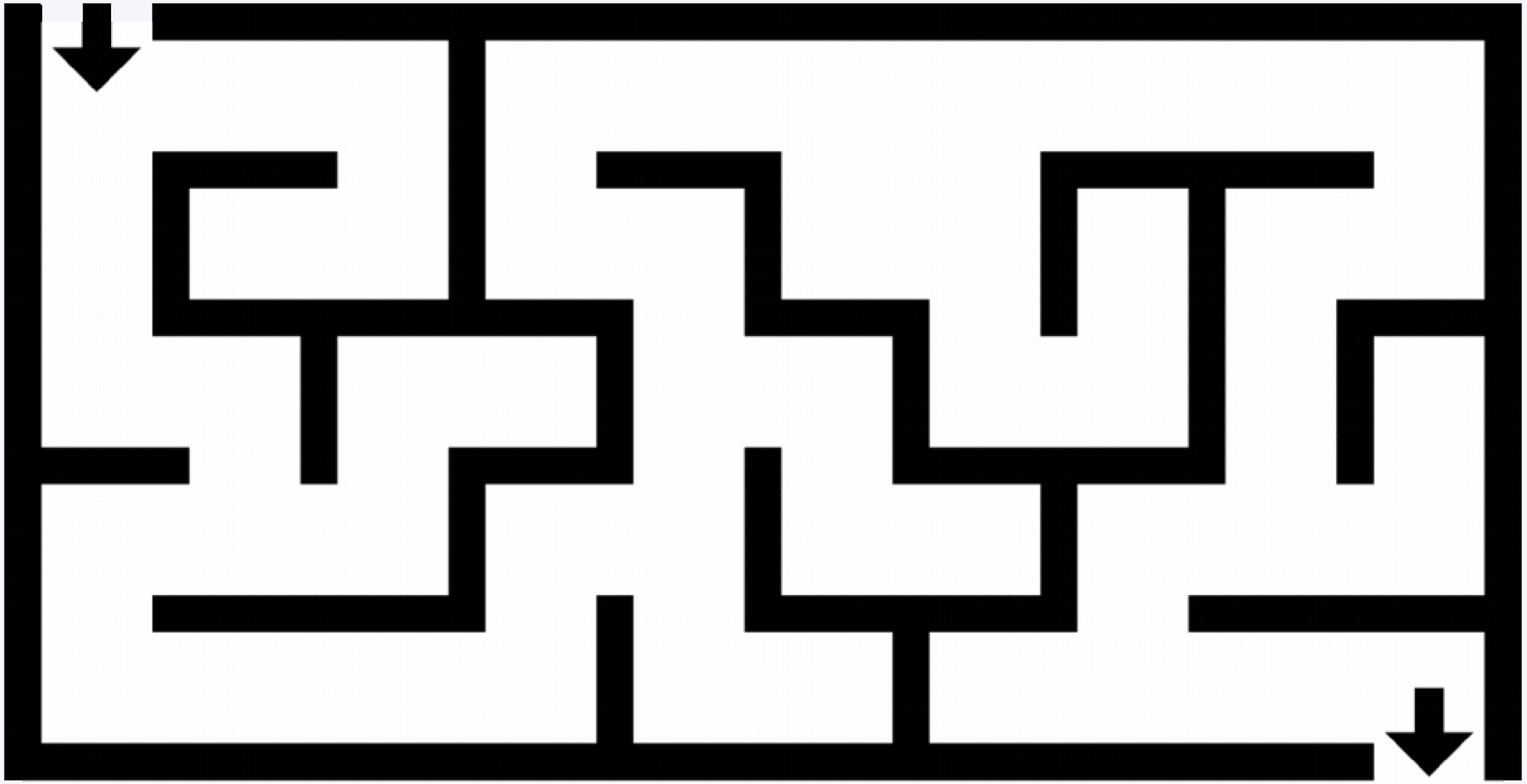
Big O of V plus E, that is vertices plus edges which is directly proportional to the size of your graph



Bit of history

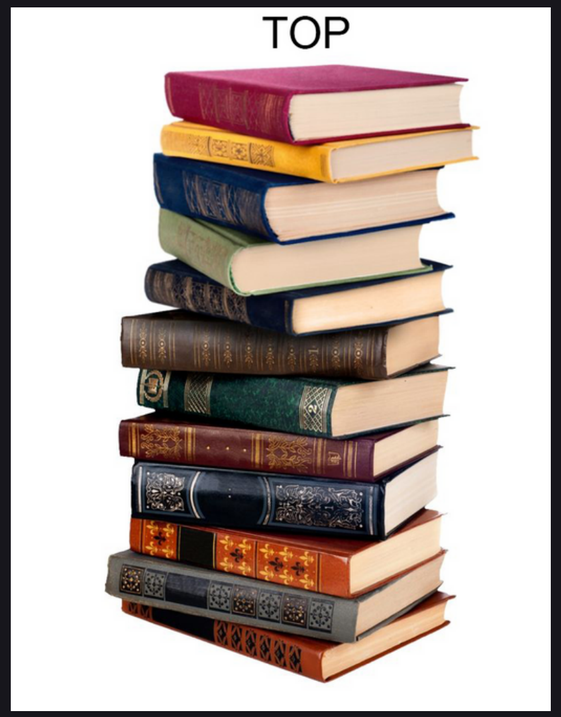
A version of depth-first search was investigated in the 19th century by French mathematician Charles Pierre Trémaux as a strategy for solving mazes

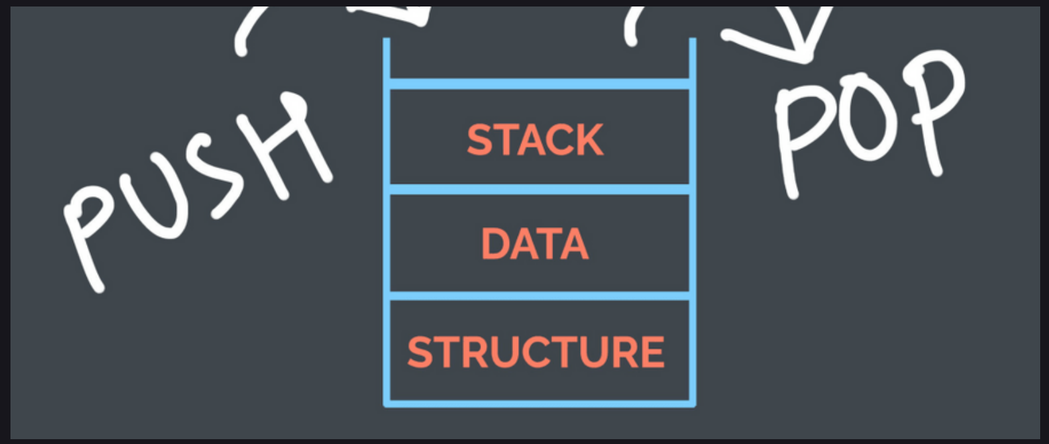




What is a Stack?

A stack is an abstract data type that is a list of elements organized according to the LIFO(Last In, First Out) principle.

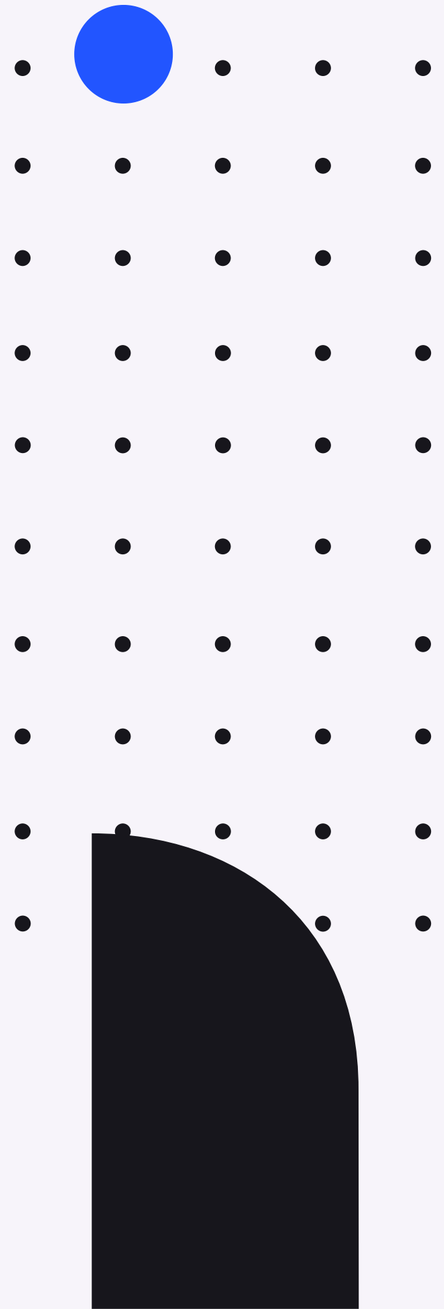




Why do I need a Stack?

Because Depth First search algorithm

- Uses a stack to remember to get the next vertex to start a search when a dead end occurs in any iteration
- Uses stack to find the shortest path



How the DFS algorithm works.

1

Start by putting any one of the graph's vertices on top of a stack.

2

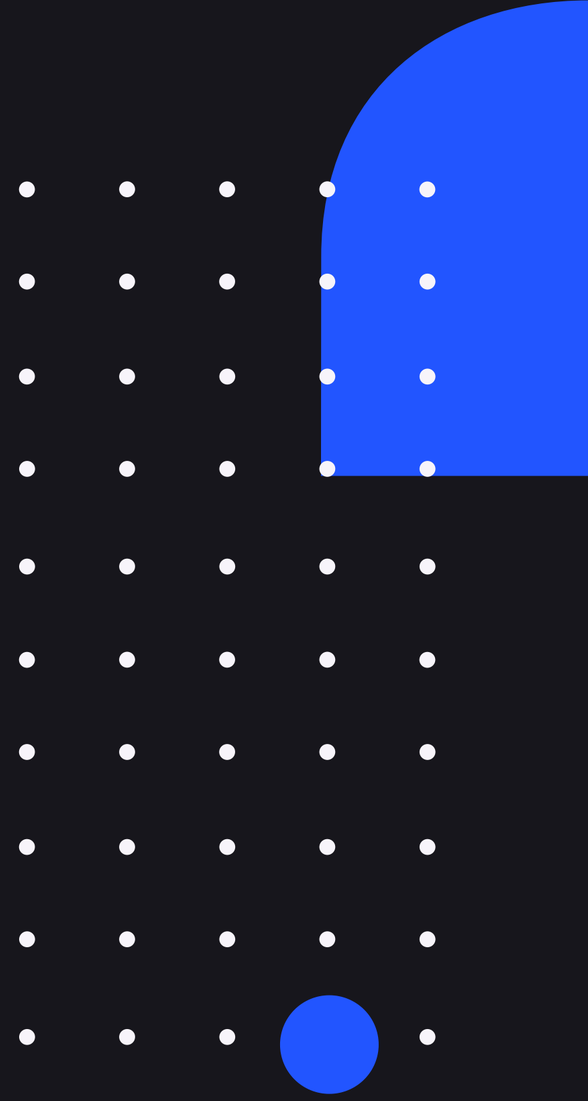
Take the top item of the stack and add it to the visited list.

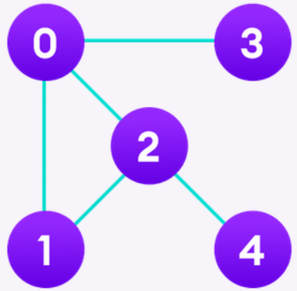
3

Create a list of that vertex's adjacent nodes. Add the ones which aren't in the visited list to the top of the stack.

4

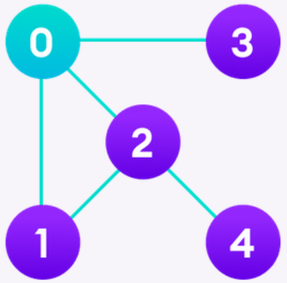
Keep repeating steps 2 and 3 until the stack is empty.





Visited

Stack

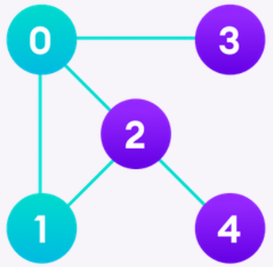


0				

Visited

1	2	3		

Stack

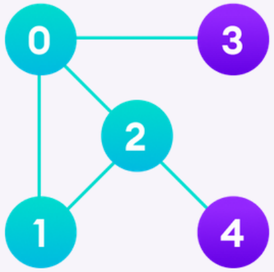


0	1			

Visited

2	3			

Stack

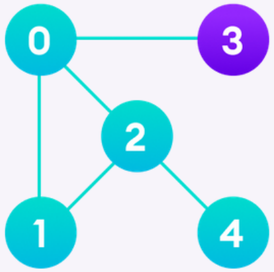


0	1	2		

Visited

4	3			

Stack

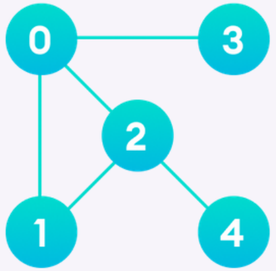


0	1	2	4	

Visited

3				

Stack



0	1	2	4	3

Visited

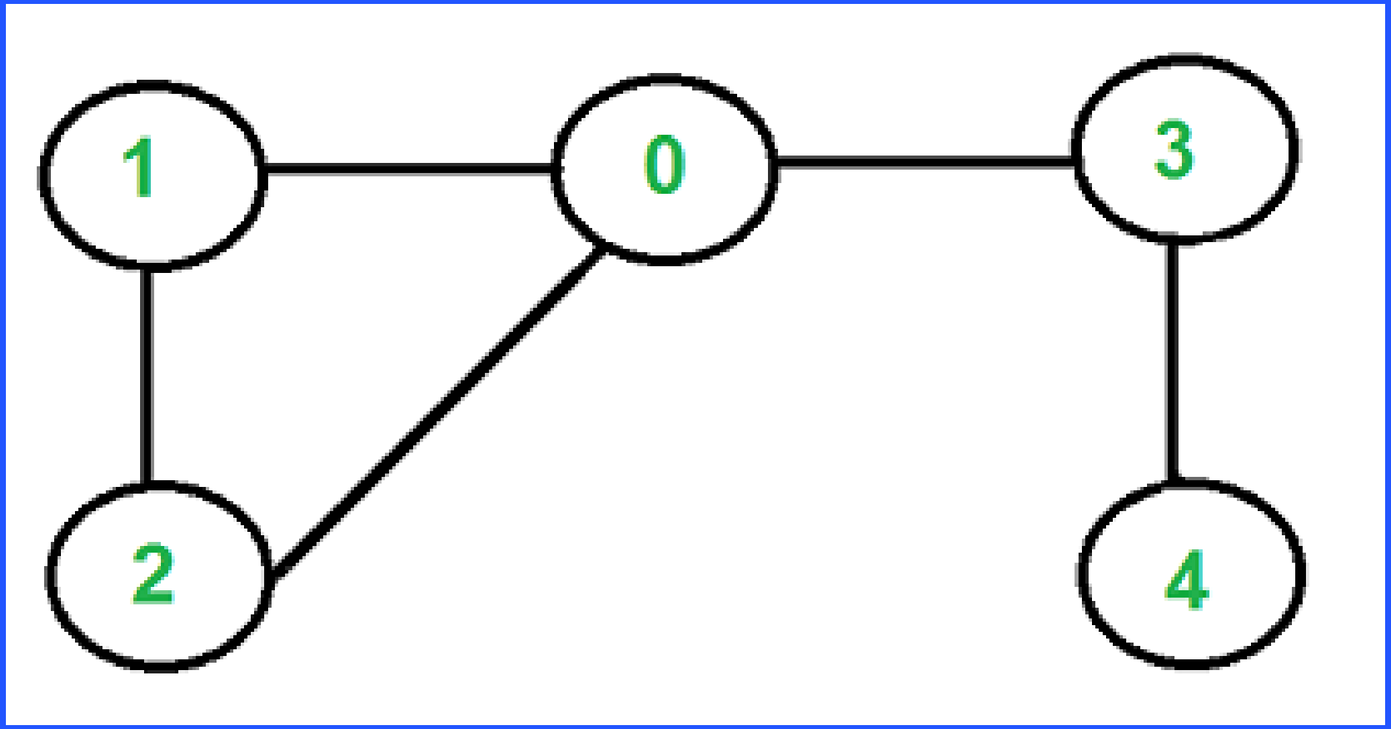
Stack

Applications

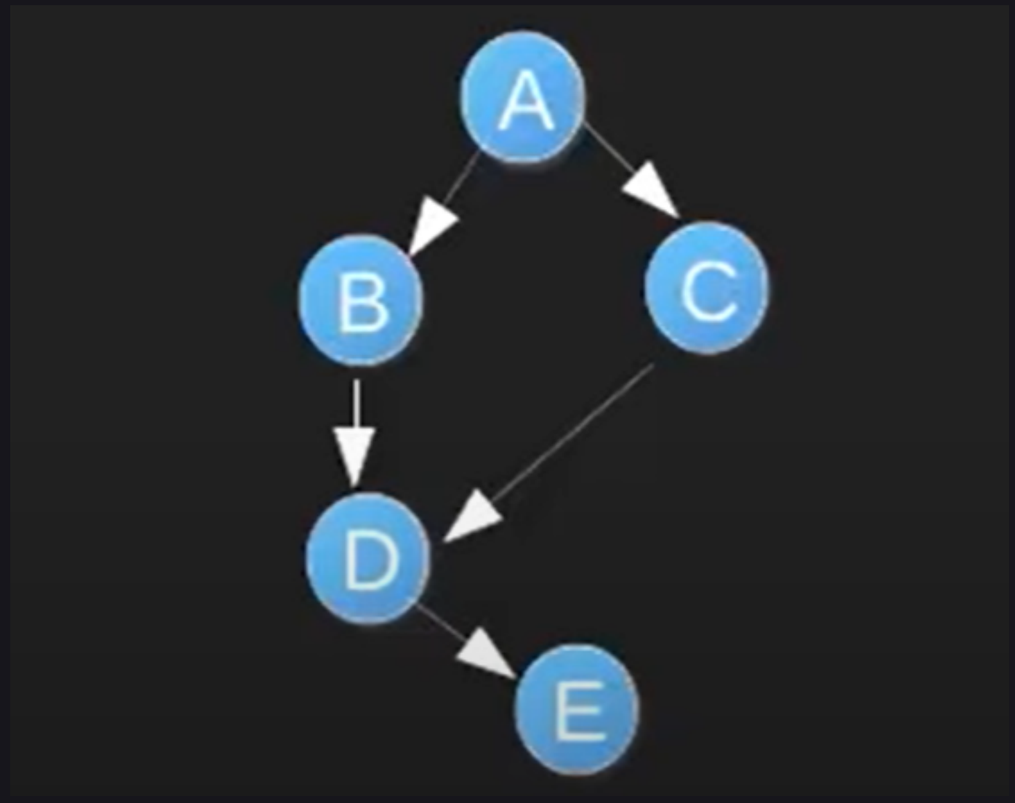
- Detecting cycle in a graph
- Topological Sorting
- Finding Strongly Connected Components
- Path Finding
- To test a graph is bipartite
- Solving puzzles with only one solution, such as mazes

Detecting a Cycle

For every visited vertex ’v’, if there is an adjacent ‘u’ such that u is already visited and u is not parent of v, then there is a cycle in graph

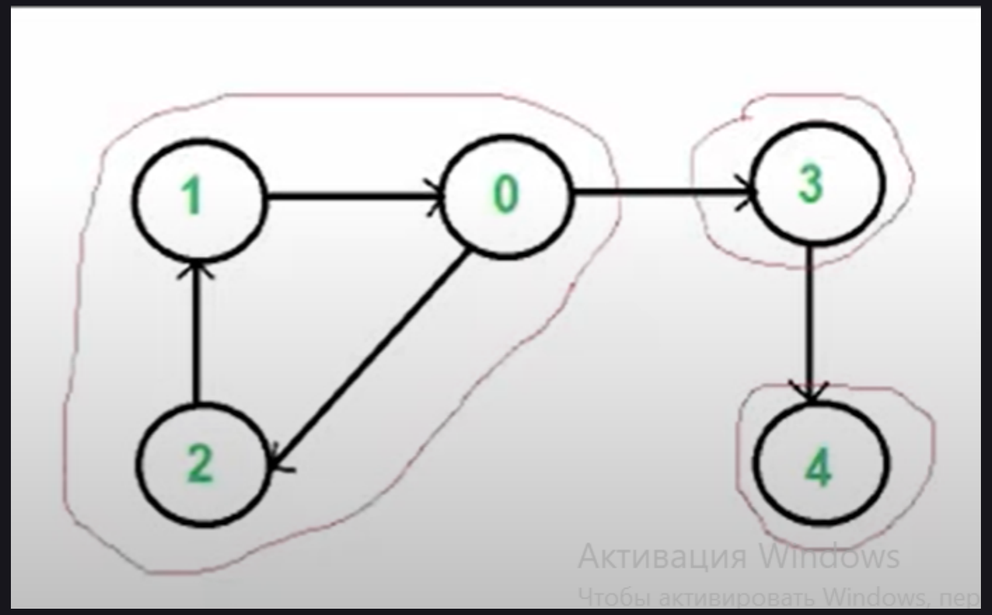


Topologica
l Sorting



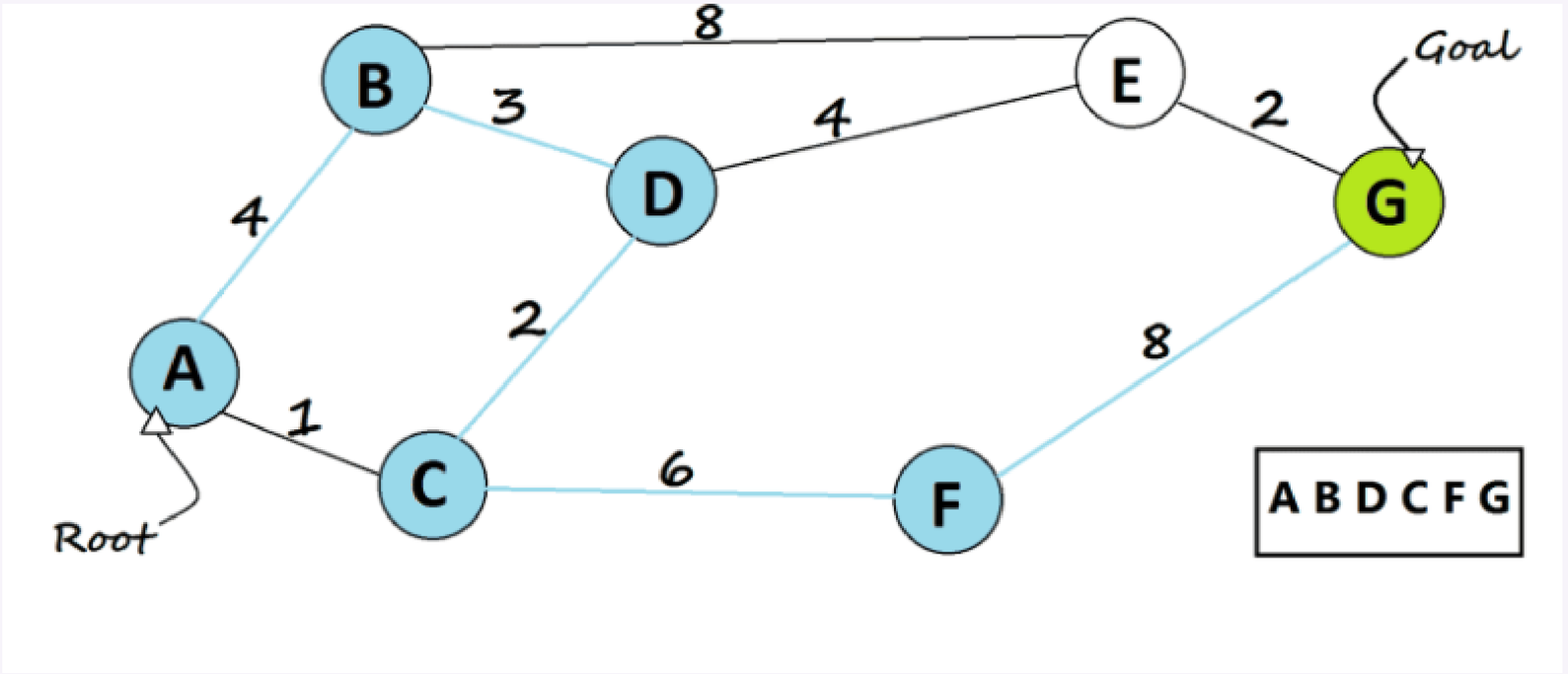
Topological sort:
A B C D E
A C B D E

Strongly Connected Components



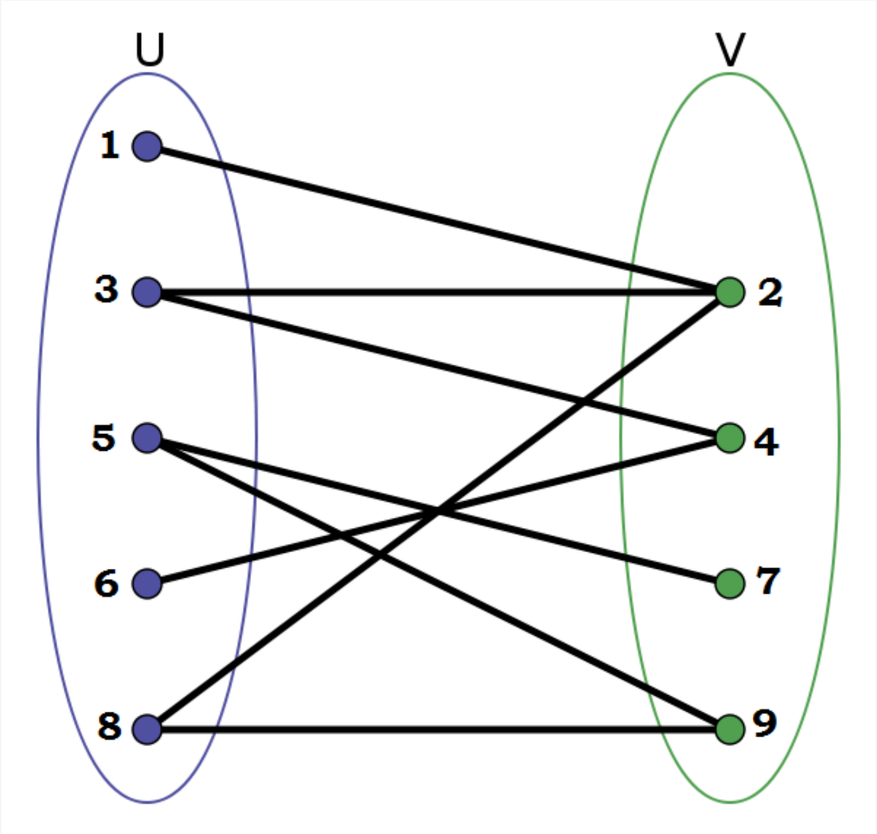
A directed graph is strongly connected if there is a path between all pairs of vertices. A strongly connected component(SCC) of a directed graph is a maximal strongly connected subgraph.

Path Finding: find a path between two given vertices u and v



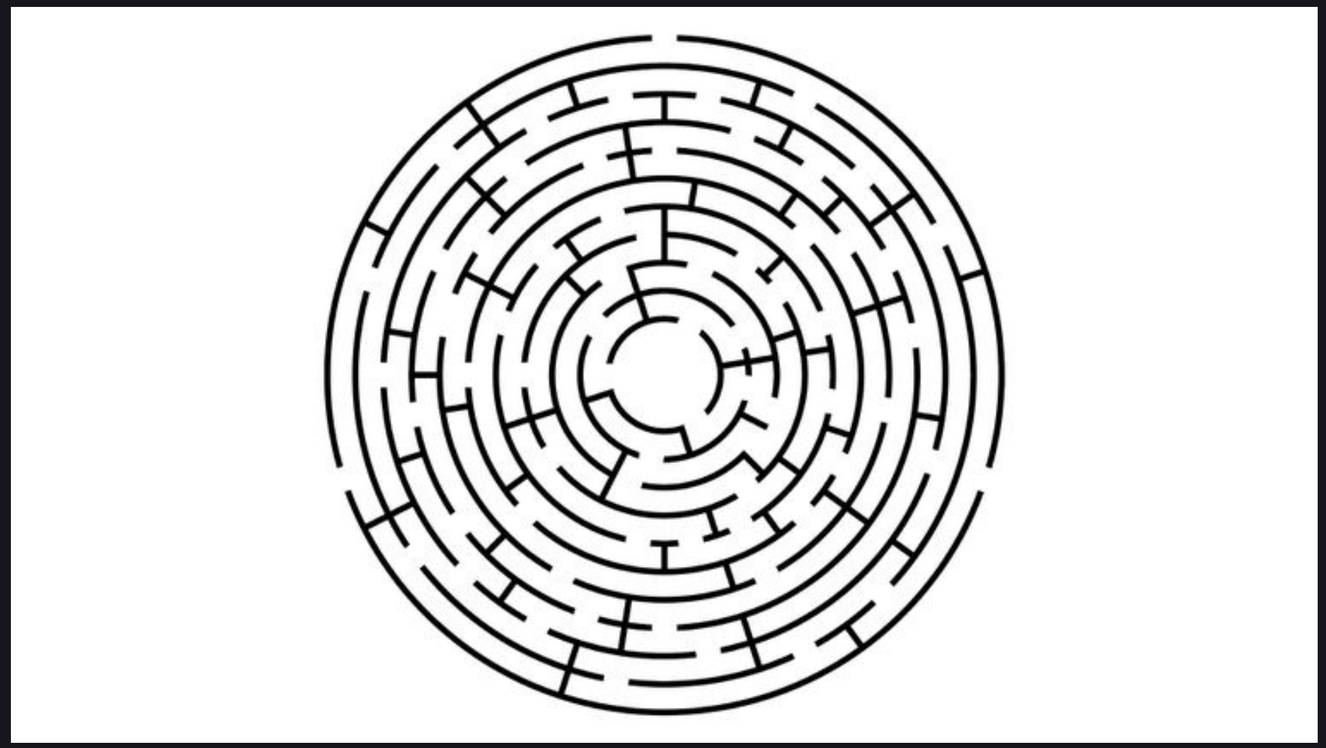
To test if a graph is bipartite

Bipartite Graph is a graph whose vertices can be divided into two independent sets, U and V such that every edge(u, v) either connects a vertex from U to V or from V to U.



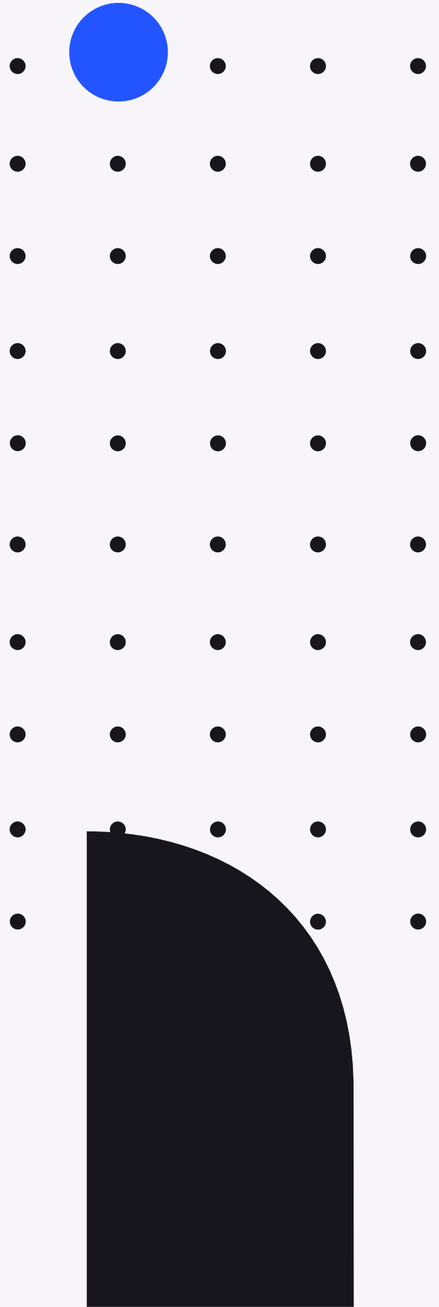
Solving puzzles problem

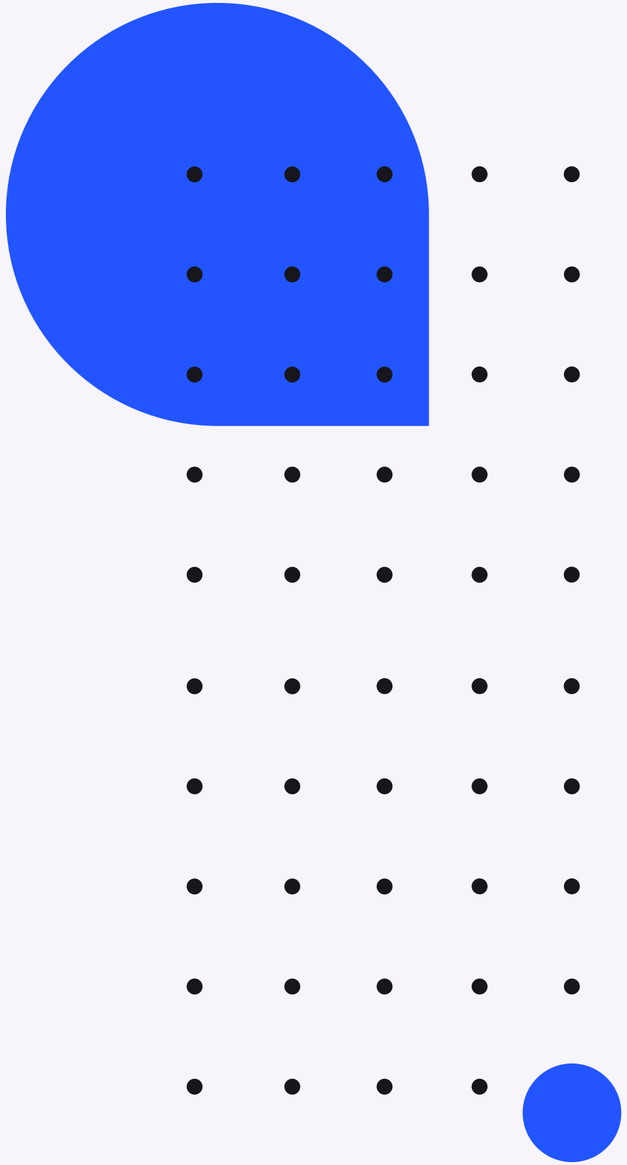
Maze related problems, obstacles in a matrix



Summary

1. We have reviewed the DFS algorithm
2. We reviewed the application
3. We tested our code and explained it





Thanks!