DEPTH-FIRST SEARCH

Advanced Algorithms

OUR TEAM

END GAME

ASSEM KOILANOVA

KUANYSHBEK BAKHTIGALIYEV

ZHANSAYA MEDEU

YERKIN SAGADILDA

DINMUKHAMED BALABEK

SUMMARY OF CONTENTS

OUR MAIN TOPICS

What is Depth First Search (DFS)?

Advantages and Disadvantages

Applications

How it works?

Example

Code implementation

WHAT IS DEPTH FIRST SEARCH (DFS)?

DFS stands for Depth First Search is a edge based technique.

It uses the Stack data structure, performs two stages, first visited vertices are pushed into stack and second if there is no vertices then visited vertices are popped.

DEPTH FIRST SEARCH

1.Depth-First Search was first investigated by French Mathematician Charles Pierre tremaux .

2.It is an algorithm for traversing tree or graph data structures .

3.One starts at the root and explores as deep as possible along each branch before backtracking .

4.It can be implemented using stack.

ADVANTAGES OF DFS

- Simple to implement;
- Needs realatively small memory for the state-space.
- The time complexity of a depth-first search to depth d is $O(b^d)$ since it generates the same set of nodes as BFS.
- If depth-first search finds solution without exploring much in a path then the time and space it takes will be very less.

DISADVANTAGES OF DFS

- Sometimes fail to find a solution(may be get stuck in an infinite long branch) - not complete;
- Not guaranteed to find an optimal solution(may not find the shortest path solution);
- Can take a lot longer to find a solution.

APPLICATIONS

- Depth First Search (DFS) searches deeper into the problem space.
- It always generates successor of the deepest unexpanded node.
- Depth First Search uses last-in first-out stack for keeping the unexpanded nodes.
- More commonly,it is implemented recursively,with the recursion stack taking the place of an explicit node stack.

HOW IT WORKS?

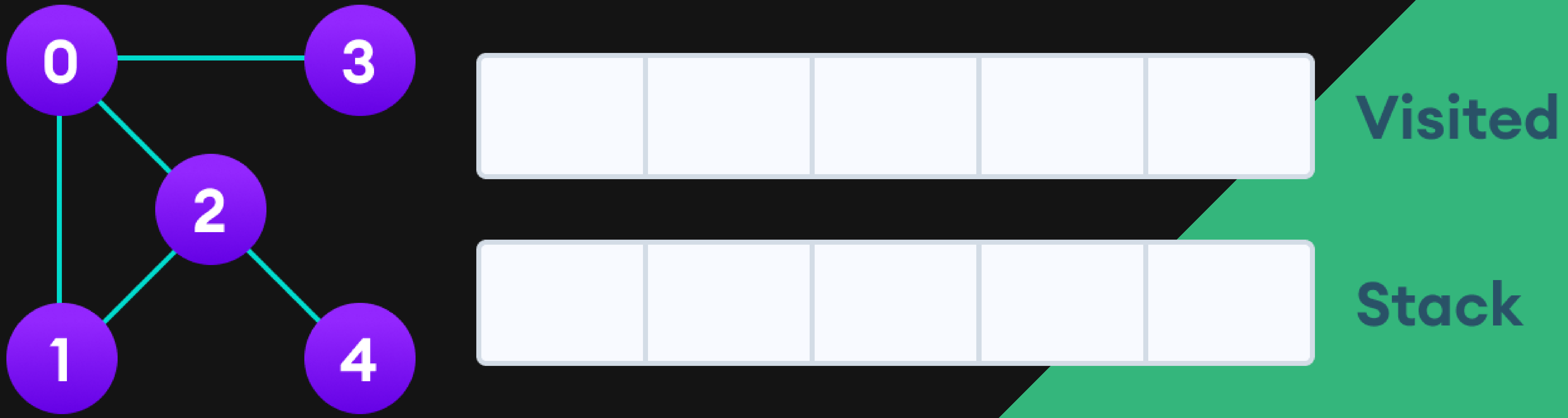
1.Start by putting any one of the graph's vertices on top of a stack.

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.

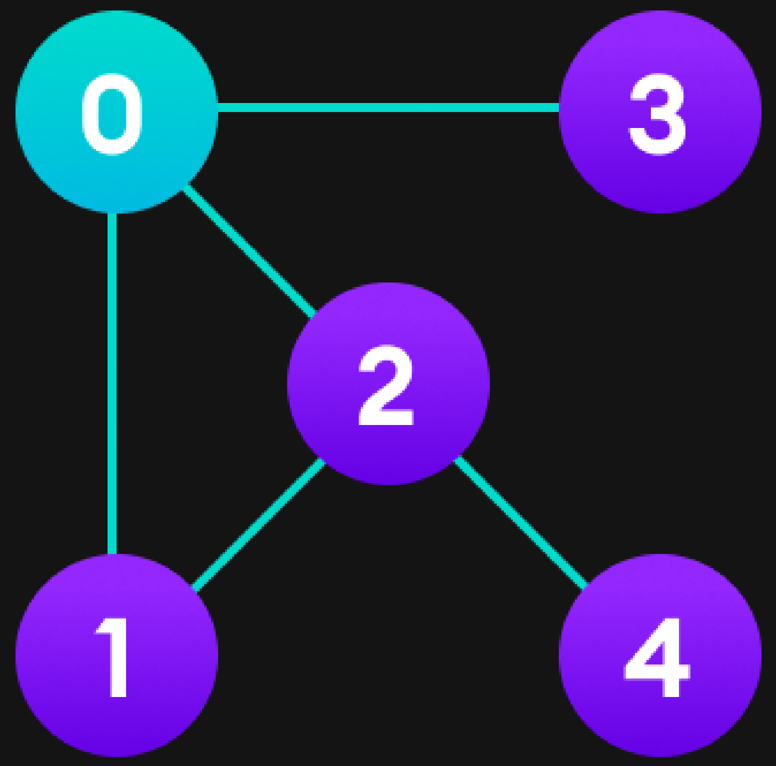
4.Keep repeating steps 2 and 3 until the stack is empty.

EXAMPLE



Undirected graph with 5 vertices

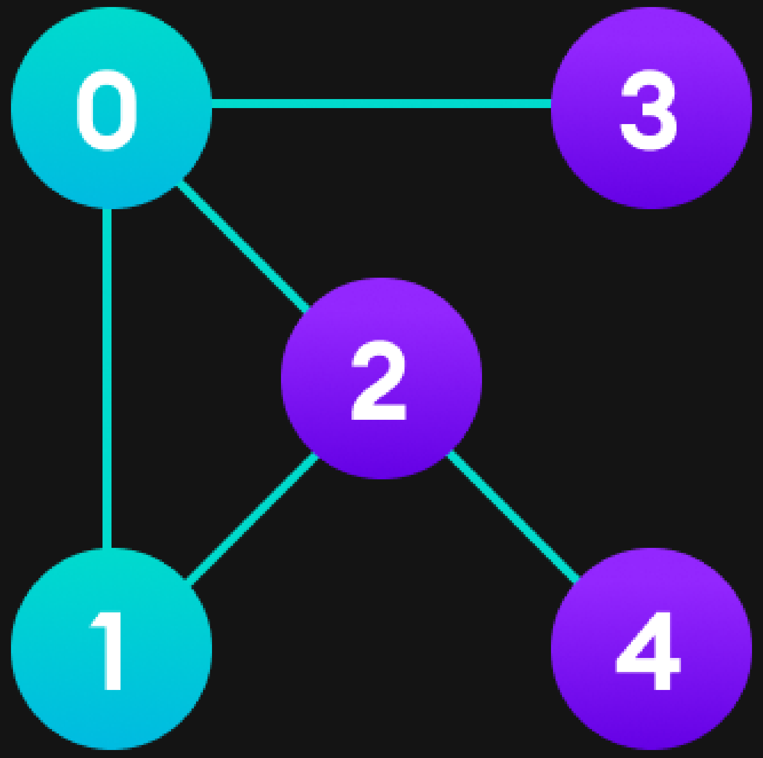
EXAMPLE



0					Visited
1	2	3			Stack

Visit the element and put it in the visited list

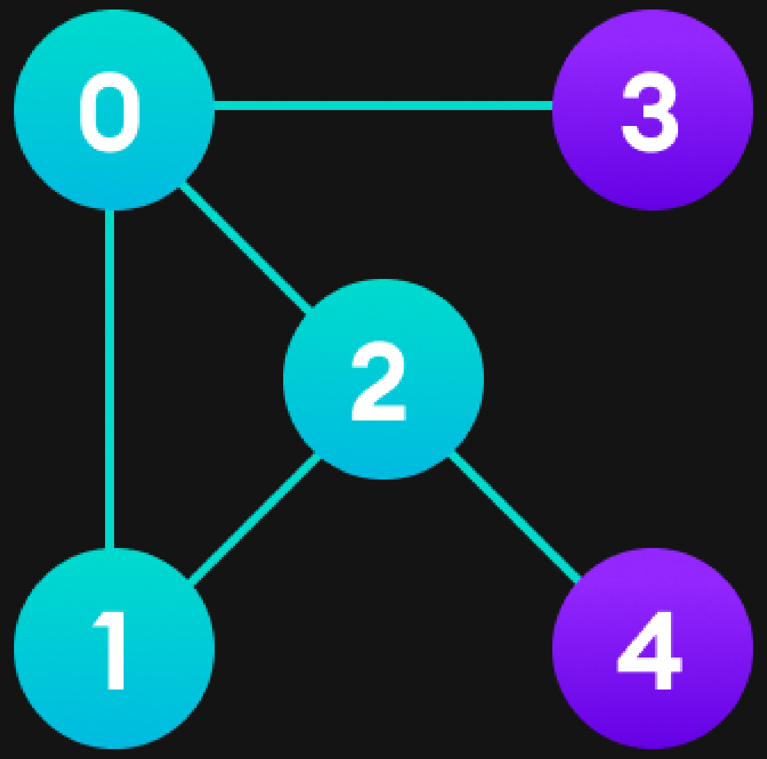
EXAMPLE



0	1				Visited
2	3				Stack

Visit the element at the top of stack

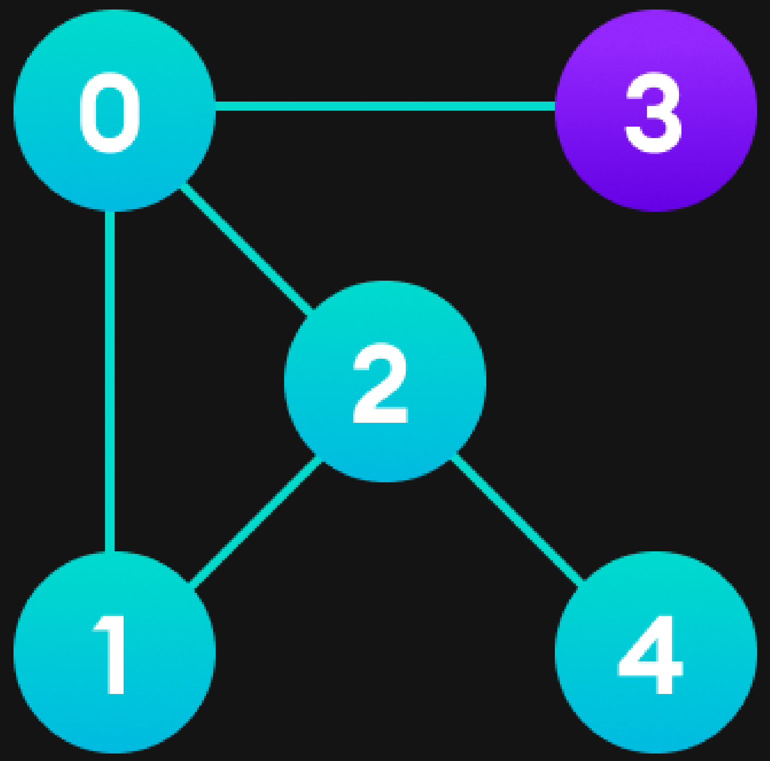
EXAMPLE



0	1	2			Visited
4	3				Stack

Vertex 2 has an unvisited adjacent vertex in 4, so we add that to the top of the stack and visit it.

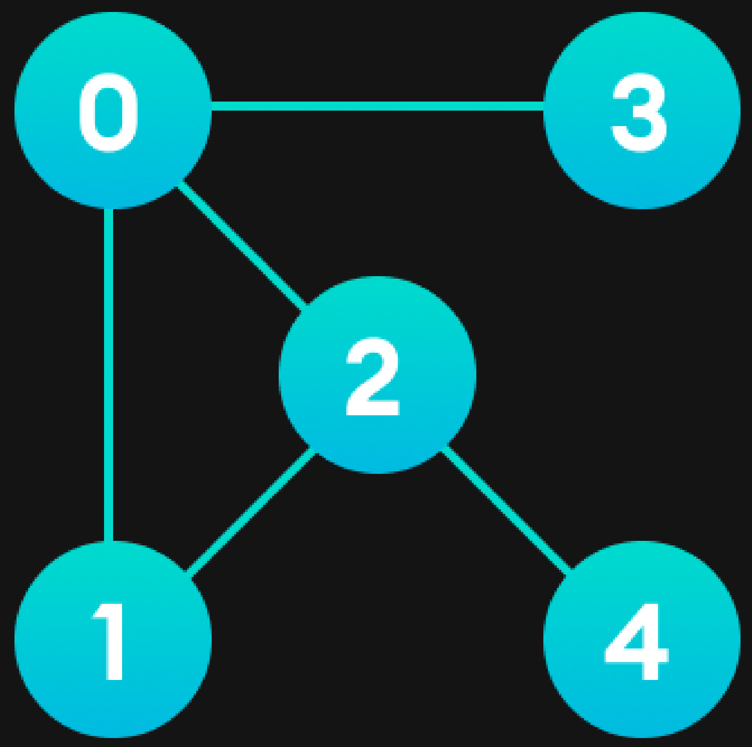
EXAMPLE



0	1	2	4		Visited
3					Stack

Vertex 2 has an unvisited adjacent vertex in 4, so we add that to the top of the stack and visit it.

EXAMPLE



0	1	2	4	3	Visited
					Stack

After we visit the last element 3, it doesn't have any unvisited adjacent nodes, so we have completed the Depth First Traversal of the graph.

CODE IMPLEMENTATION IN JAVA

THANK YOU