

**JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY**

**DEPARTMENT OF CSE & IT**

**MINI PROJECT**

**(DATA STRUCTURES)**

**Graph Generics**

**GROUP MEMBERS:**

**RIYA SAXENA 19103215 B7**

**KARTIKE TIWARI 19103216 B7**

**ABHINAY SINGH 19103226 B7**

**ABHISHEK ANAND 19103236 B7**

**ABSTRACT OF PROJECT:**

**Graphs:**

A Graph is a non-linear data structure consisting of nodes and edges. The nodes are sometimes also referred to as vertices and the edges are lines or arcs that connect any two nodes in the graph.

Graphs are used to solve many real-life problems. Graphs are used to represent networks. The networks may include paths in a city or telephone network or circuit network. Graphs are also used in social networks like LinkedIn, Facebook.

**Graphs Generics:**

Generics is the idea to allow type (Integer, String, etc. and user-defined types) to be a parameter to methods, classes and interfaces. For example, classes like an array, map, etc., which can be used using generics very efficiently. We can use them for any type.

The method of Generic Programming is implemented to increase the efficiency of the code. Generic Programming enables the programmer to write a general algorithm which will work with all data types. It eliminates the need to create different algorithms if the data type is an integer, string or a character.

**Algorithms to include in our Graph Generics:**

1. Depth First Search (DFS)
2. Breadth First Search (BFS)
3. Minimum Spanning Tree
4. Cycle Detection
5. Disjoint Set Union
6. Dijkstra’s Algorithm
7. Topological Sort (Kahn’s Algorithm)
8. Kosaraju’s Algorithm for SCC
9. Bipartite Check of a Graph
10. Longest path in directed acyclic graph (Dynamic Programming)

The Advantages of Generic Programming are:

* *Code Reusability*

Avoid Function Overloading

Once written it can be used for multiple times and cases.

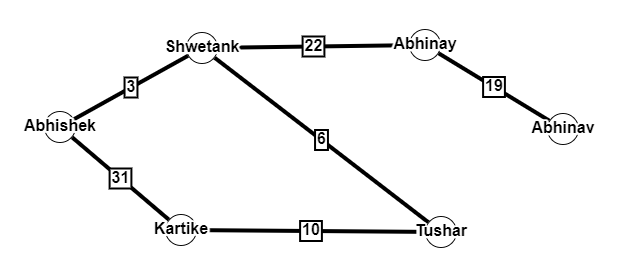
Generics can be implemented in C++ using Templates. Template is a simple and yet very powerful tool in C++. The simple idea is to pass data type as a parameter so that we don’t need to write the same code for different data types. For example, a software company may need sort() for different data types. Rather than writing and maintaining the multiple codes, we can write one sort() and pass data type as a parameter.

* *Generic Functions using Template:*

We write a generic function that can be used for different data types. Examples of function templates are sort(), max(), min(), printArray() .

**DATASET DESCRIPTION:**

**Undirected Graph G (Weighted) Description:**



Node (Shwetank) is connected to Node (Abhinay)

Node (Shwetank) is connected to Node (Abhishek)

Node (Kartike) is connected to Node (Tushar)

Node (Tushar) is connected to Node (Shwetank)

Node (Kartike) is connected to Node (Abhishek)

Node (Abhinay) is connected to Node (Abhinav)

(Shwetank) is located at a distance of 22 from (Abhinay)

(Shwetank) is located at a distance of 3 from (Abhishek)

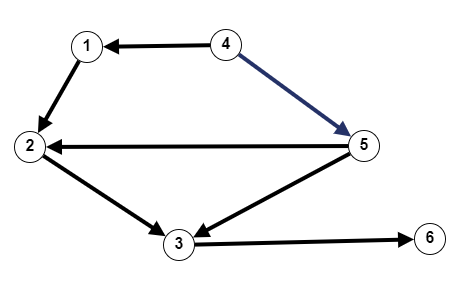
(Kartike) is located at a distance of 10 from (Tushar)

(Tushar) is located at a distance of 6 from (Shwetank)

(Kartike) is located at a distance of 31 from (Abhishek)

(Abhinay) is located at a distance of 19 from (Abhinav)

**Directed Graph G1 (Unweighted) Description:**



Node (1) is connected to Node (2)

Node (2) is connected to Node (3)

Node (3) is connected to Node (6)

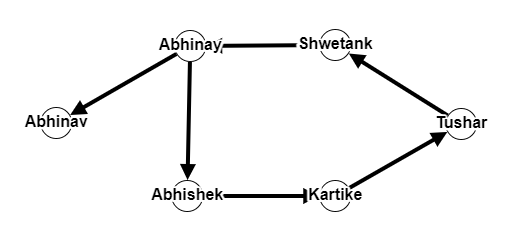
Node (4) is connected to Node (1)

Node (4) is connected to Node (5)

Node (5) is connected to Node (6)

Node (5) is connected to Node (2)

**Directed Graph G2 (Unweighted) Description:**



Node (Shwetank) is connected to Node (Abhinay)

Node (Abhinay) is connected to Node (Tushar)

Node (Tushar) is connected to Node (Shwetank)

Node (Kartike) is connected to Node (Tushar)

Node (Abhishek) is connected to Node (Kartike)

Node (Abhinay) is connected to Node (Abhinav)

# **DESIGN OF THE PROJECT**

The idea of the project is to implement Generics on Graphs. In order to implement Graph in easier way we have created a template that uses

Concept of Object Oriented Programming, Array of Vector, Array, Queue.

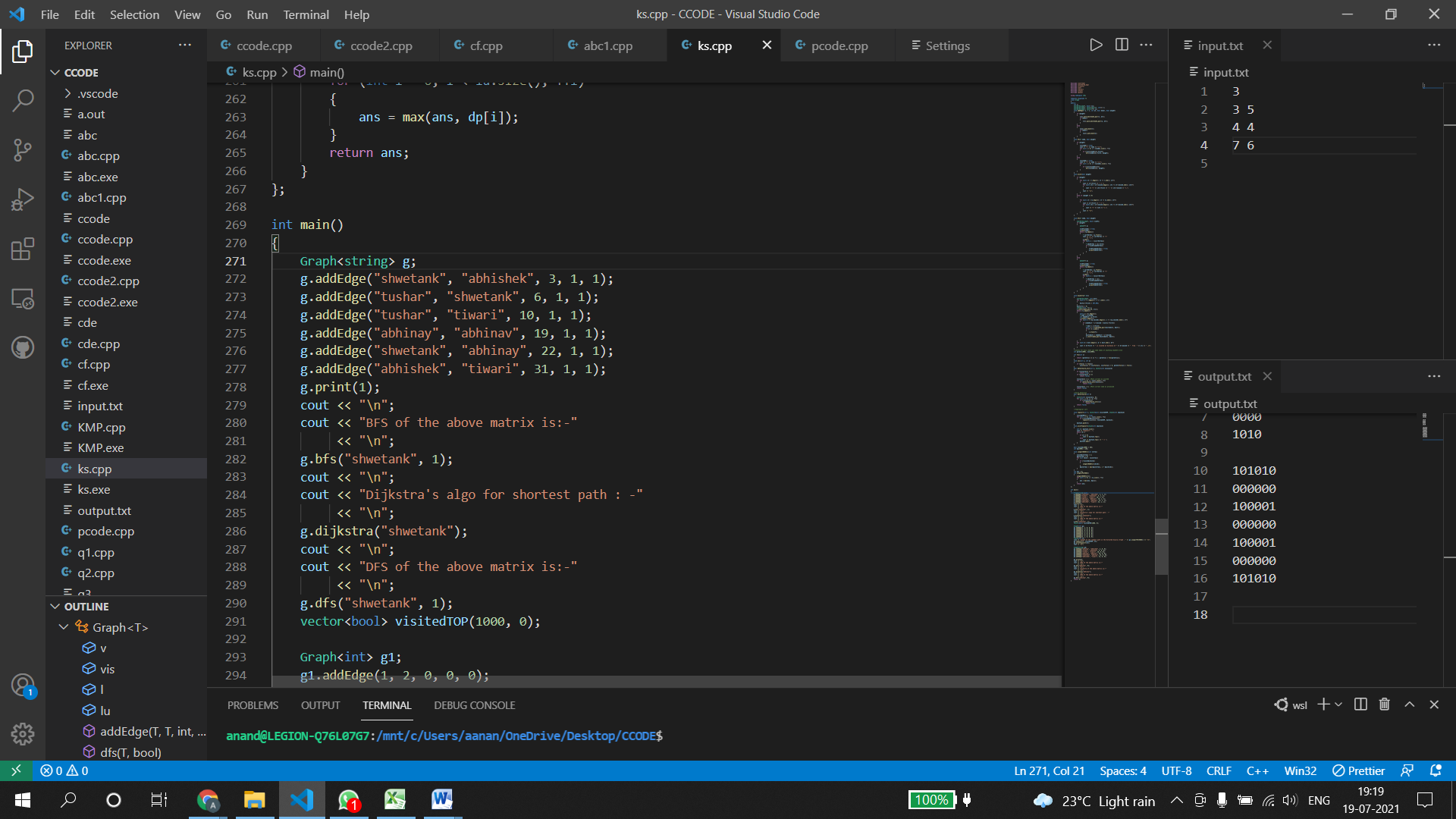
We have created a class with the name of Graph where we have declared and initialised different types of graphs. Then we have further used OOPS, to create member functions in the same class, so that we can implement the different algorithms with ease.

Then we have implemented various Graph Algorithms on different types of graphs.

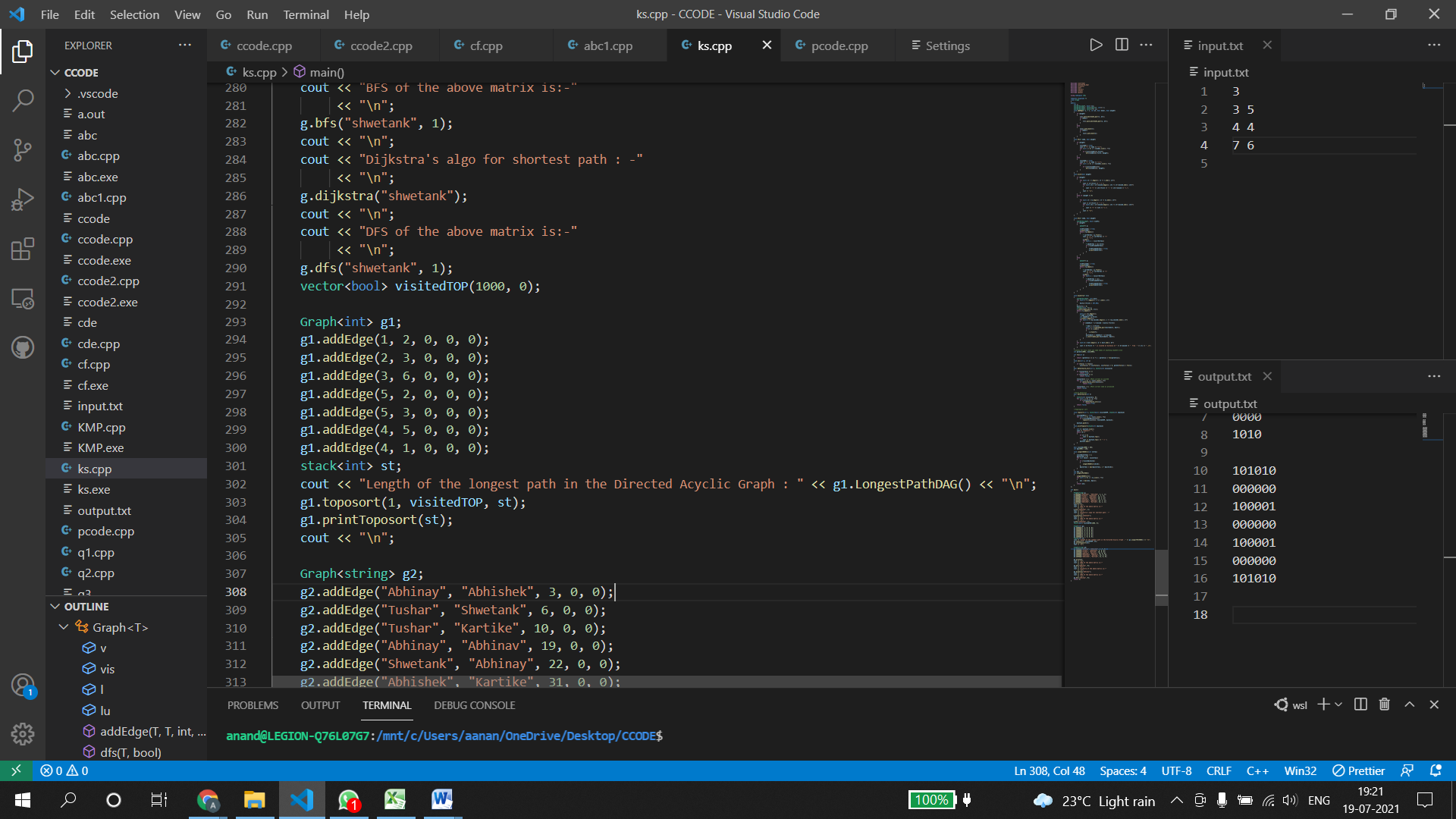
Some of those Algorithms are DFS,BFS, Dijkstra’s Algorithms, etc.

First, we have created 3 different types of graphs using Adjacency List (array of vectors).

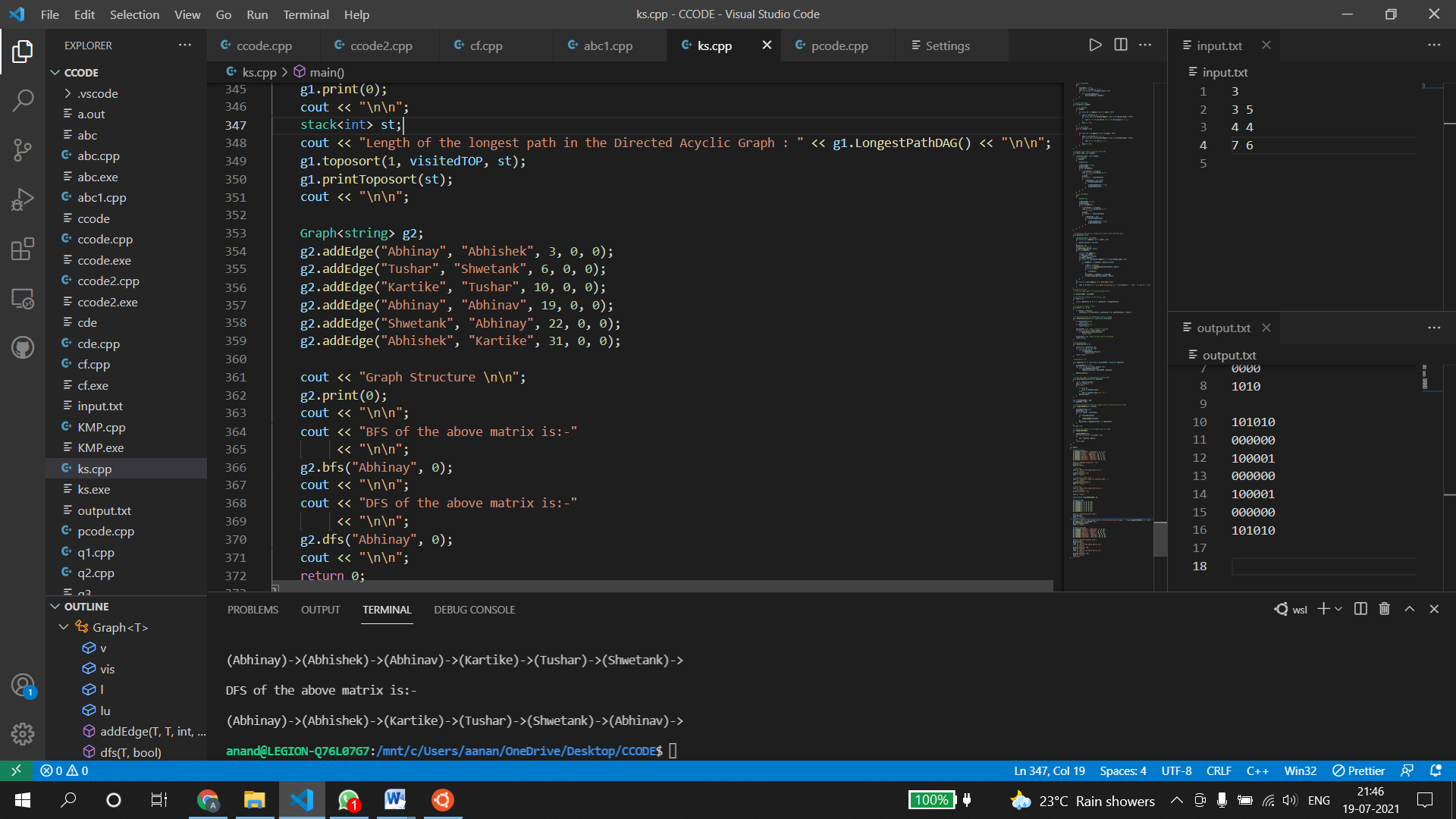
**Graph 1:**



**Graph 2:**



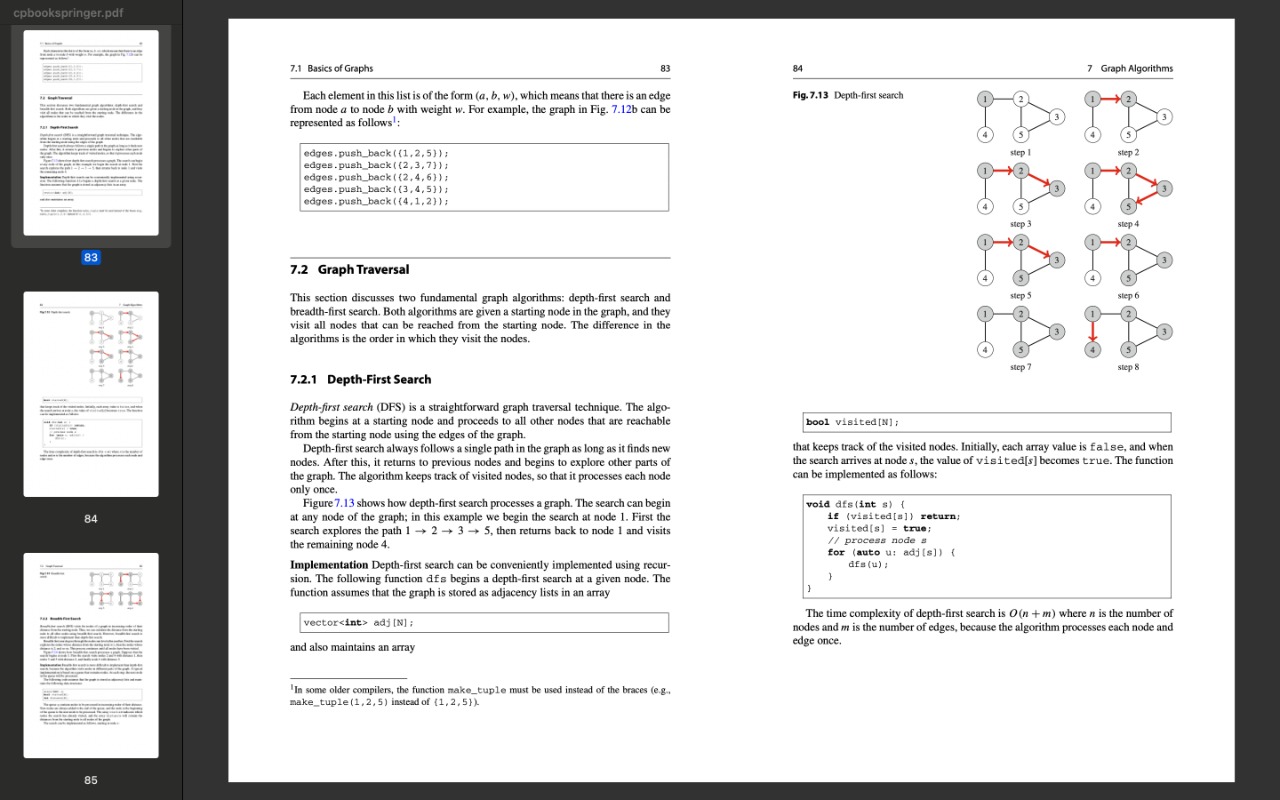
**Graph 3:**



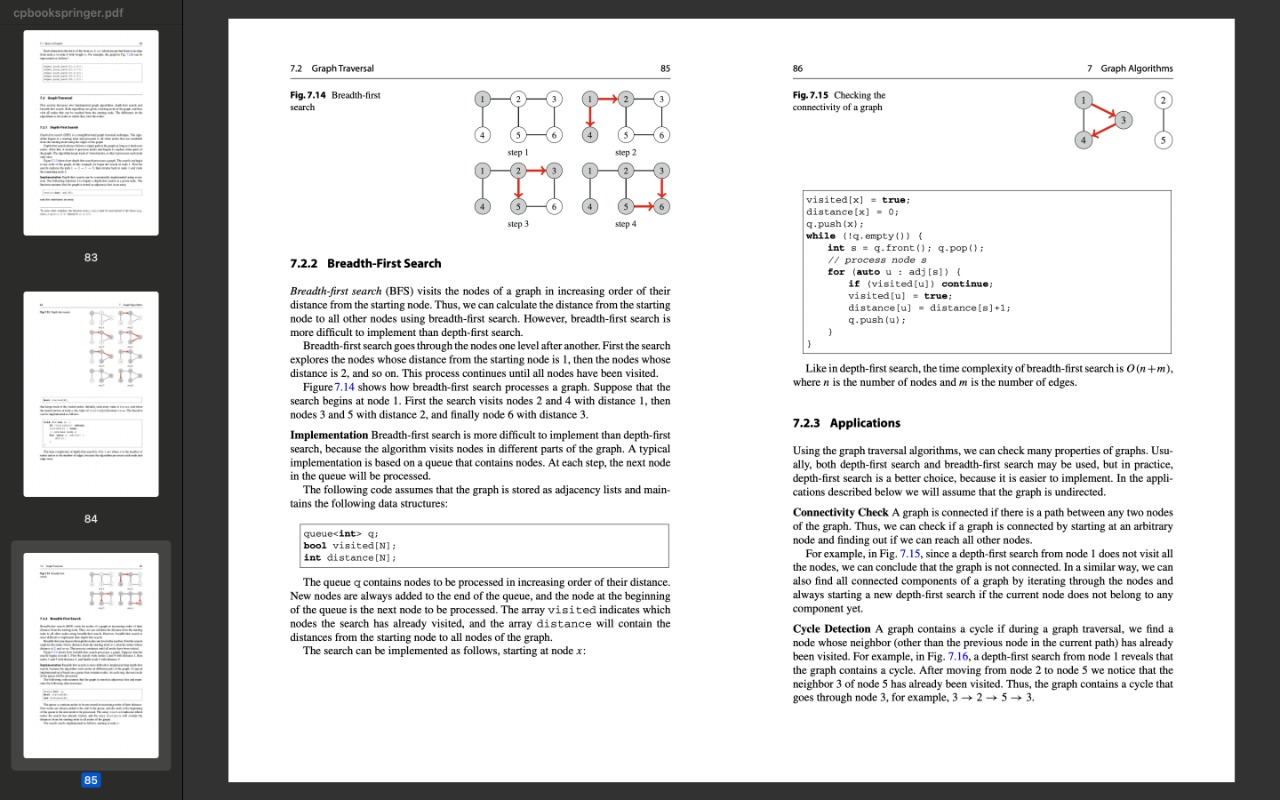
Then we have implemented some basic Graph Algorithms such as DFS, BFS and common questions like Cycle Detection and Topological Sort.

**Algorithms we have used:**

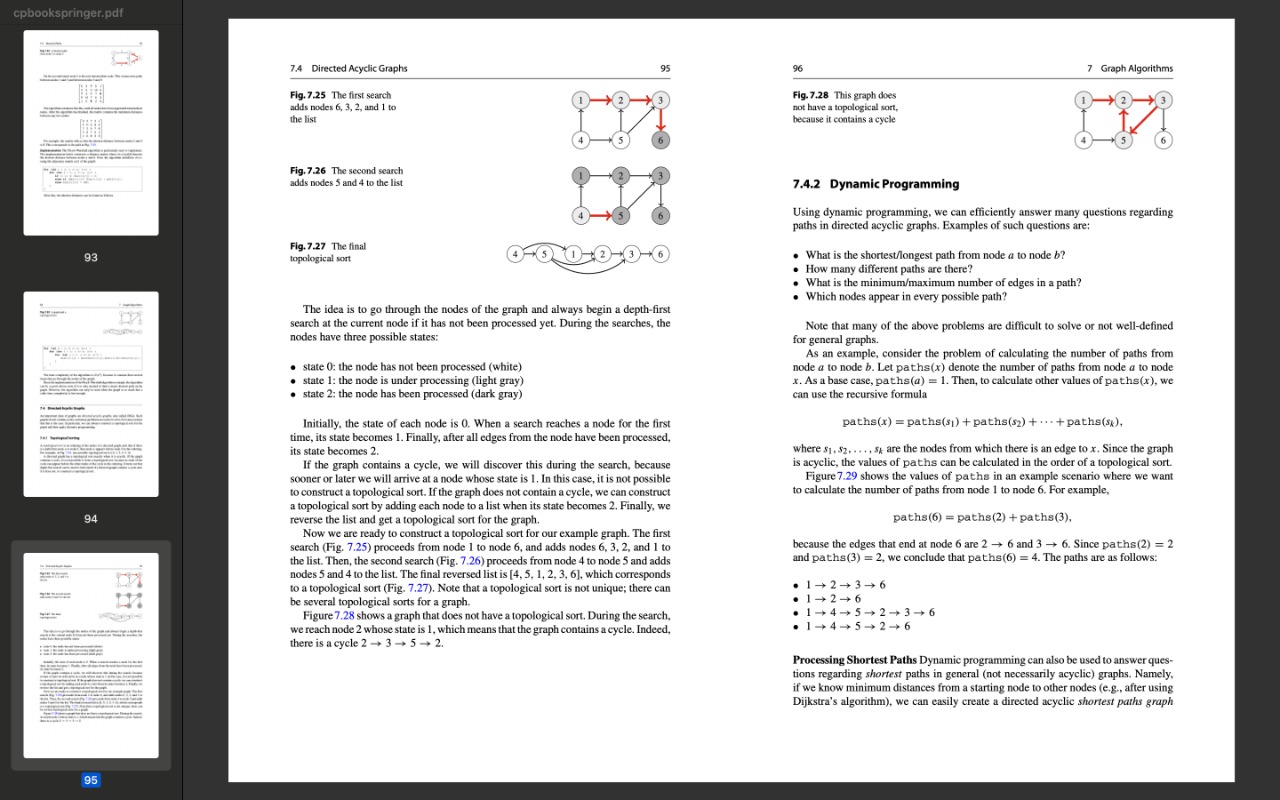
**Depth-first search (DFS)** is an algorithm for traversing or searching graph data structures. Th a elgorithm starts at the root node and explores as far as possible along each branch before backtracking.



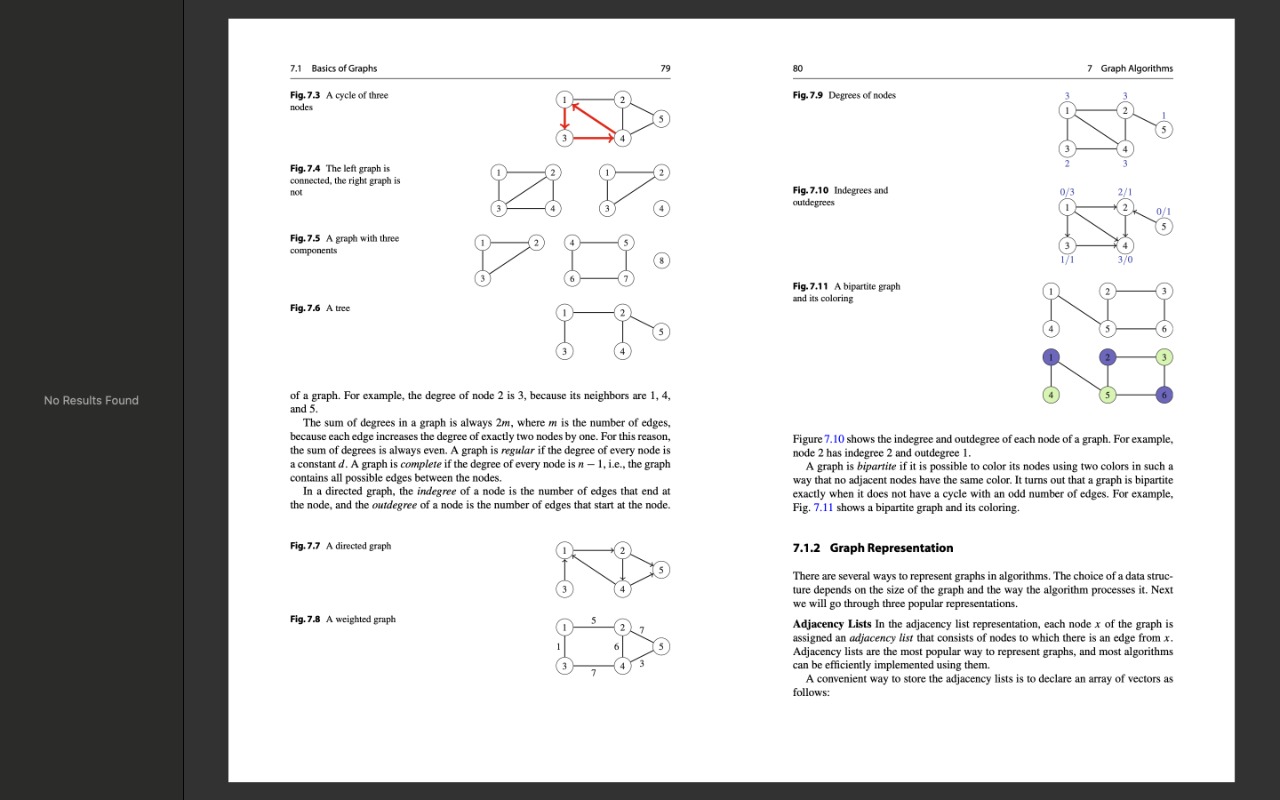
**Breadth first search** **(BFS)** is a graph traversal algorithm that starts traversing the graph from root node and explores all the neighbouring nodes.



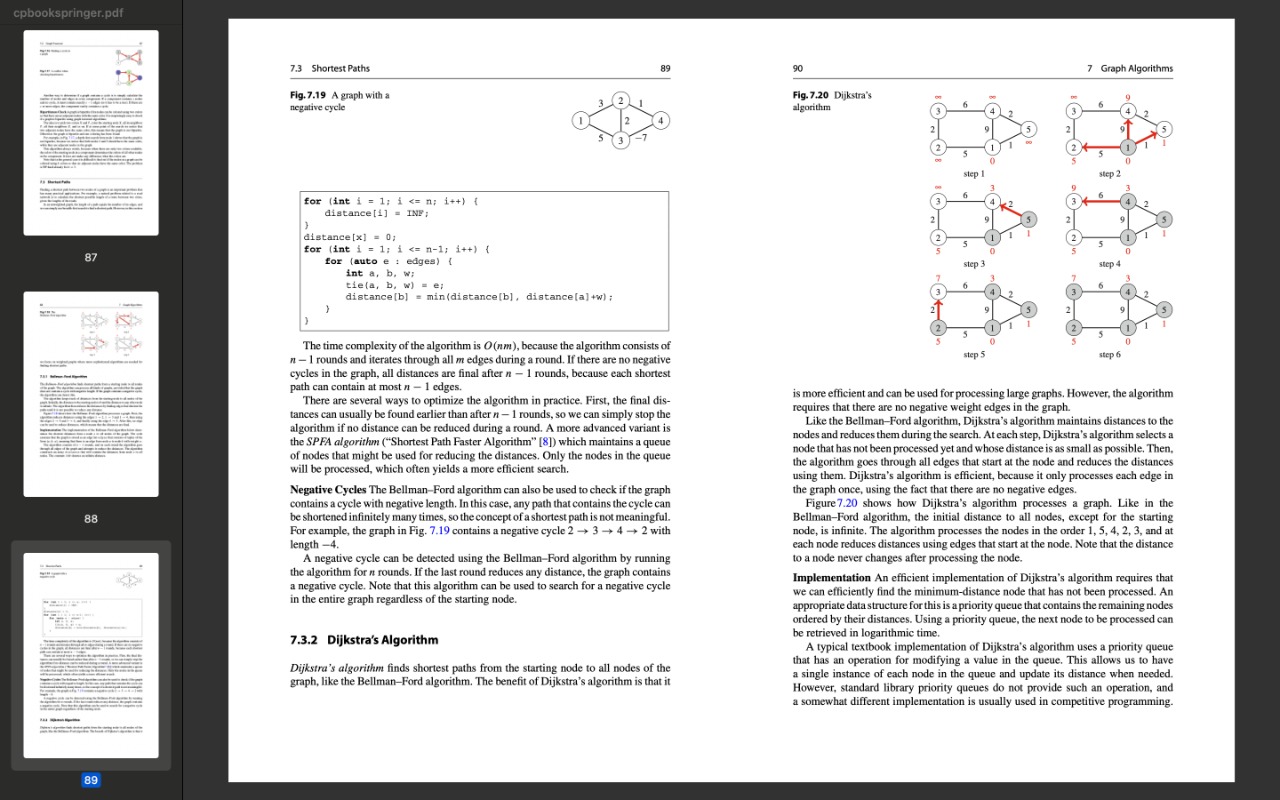
**Topological sort:** It is an ordering of the nodes of a directed graph such that if there is a path from node a to node b, then node a appears before node b in the ordering.one possible topological sort is [4, 1, 5, 2, 3, 6].



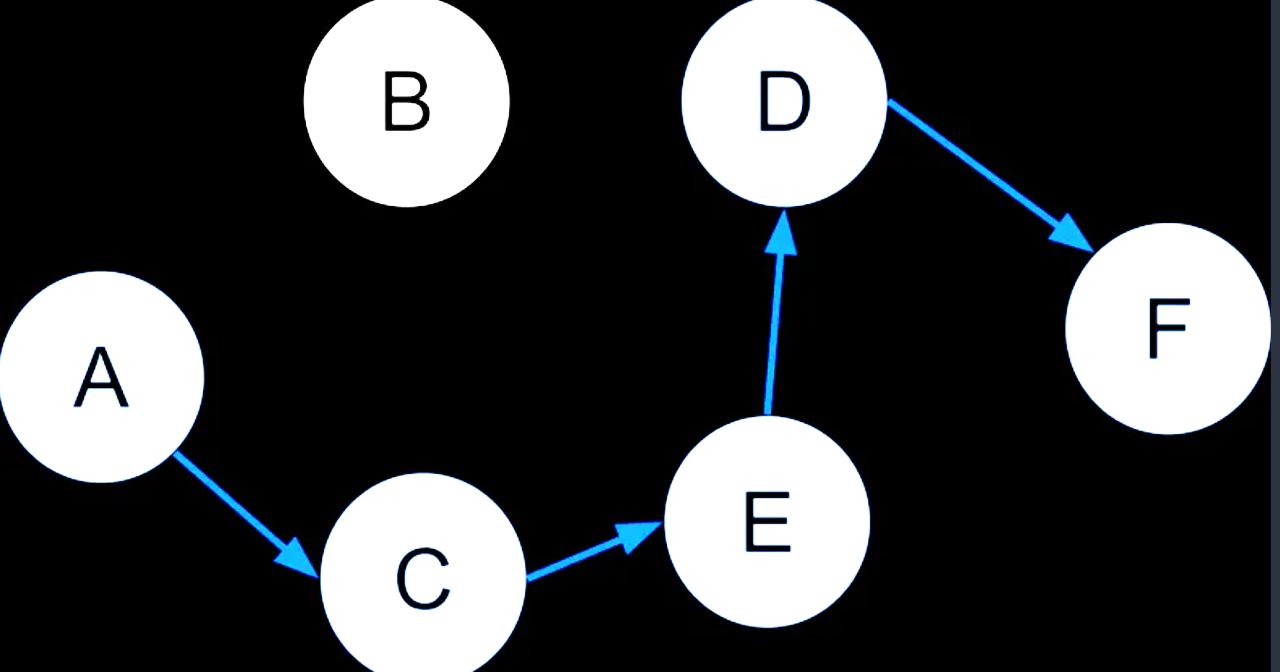
**Detect Cycle:** Written an algorithm to check if there is a cycle or not in the graph.



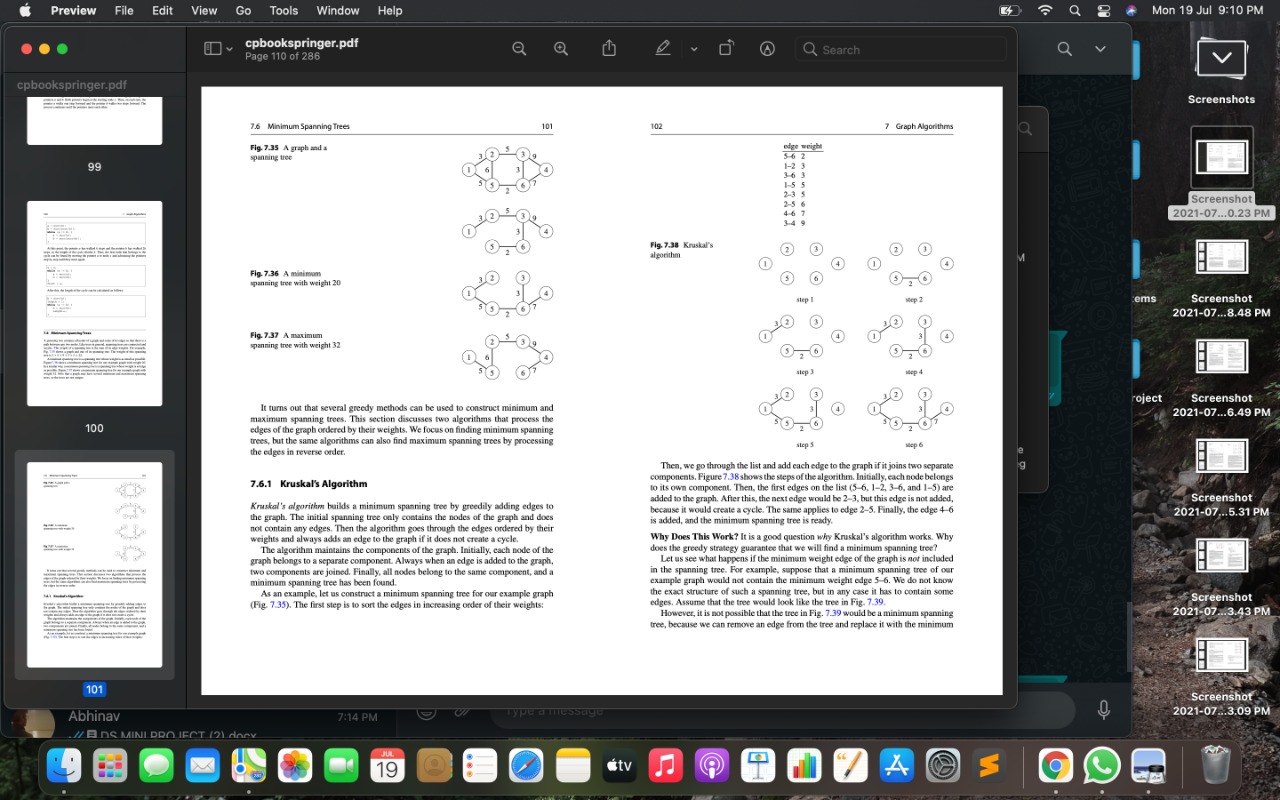
**Dijkstra's** algorithm is an algorithm for finding the shortest paths between nodes in a graph



**LONGEST PATH IN A DAG:** Written an algorithm to find the length of the longest path in a DAG with the help of Dynamic Programming.



**Kruskal**'s algorithm finds a minimum spanning tree of an undirected edge-weighted graph. If the graph is connected, it finds a minimum spanning tree.



**DATA STRUCTURES USED:**

1. Graph – Adjacency List Representation
2. Queue Data Structure -

In this project bfs algorithm is used, which requires queue data structure, to traverse the adjacency list.

1. Stack

Used in Topological Sort

1. Linked List
2. Array

**LANGUAGE USED:**

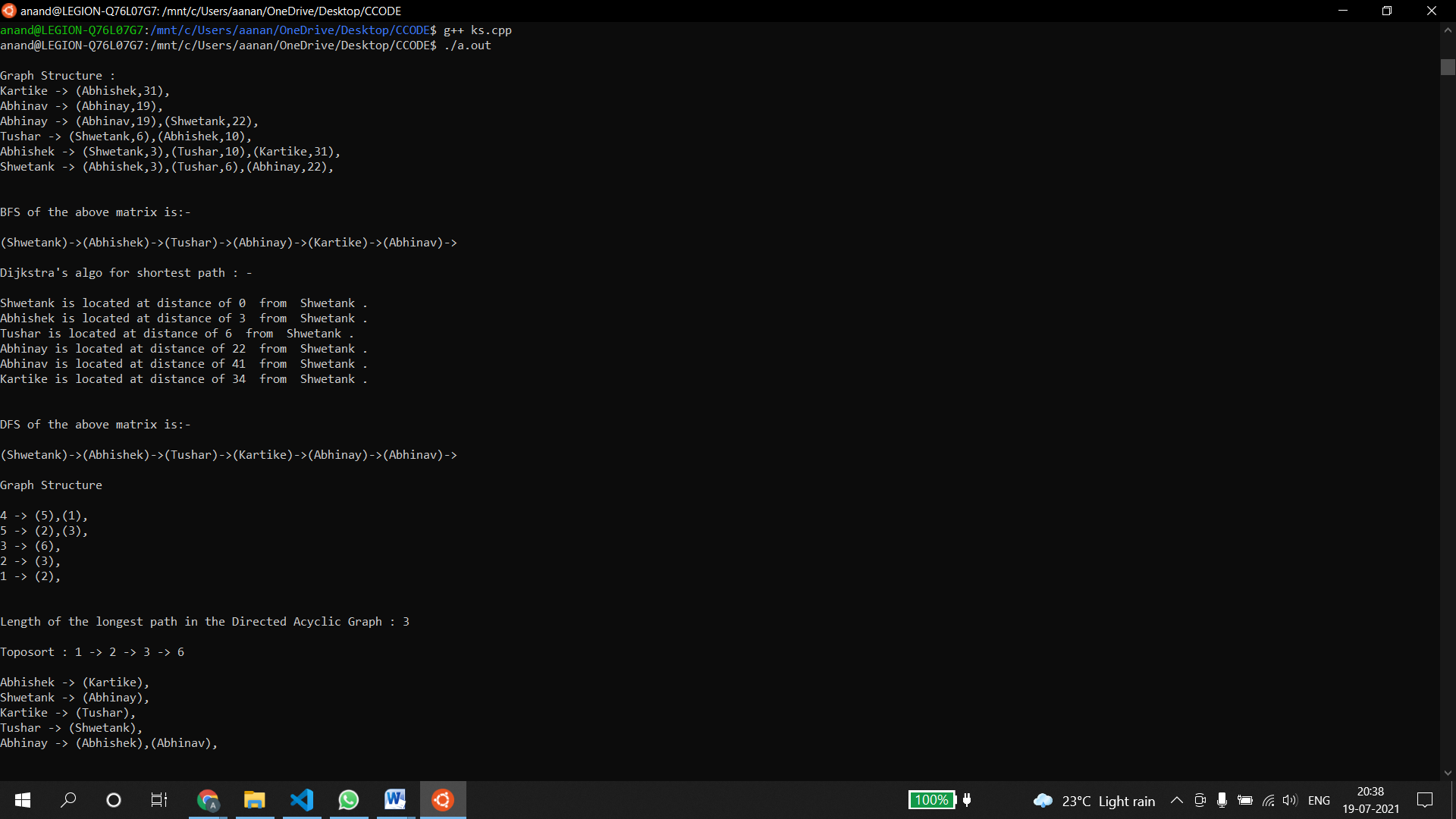
In this project C++ language is used.

**TOOLS USED:**

VISUAL STUDIO CODE

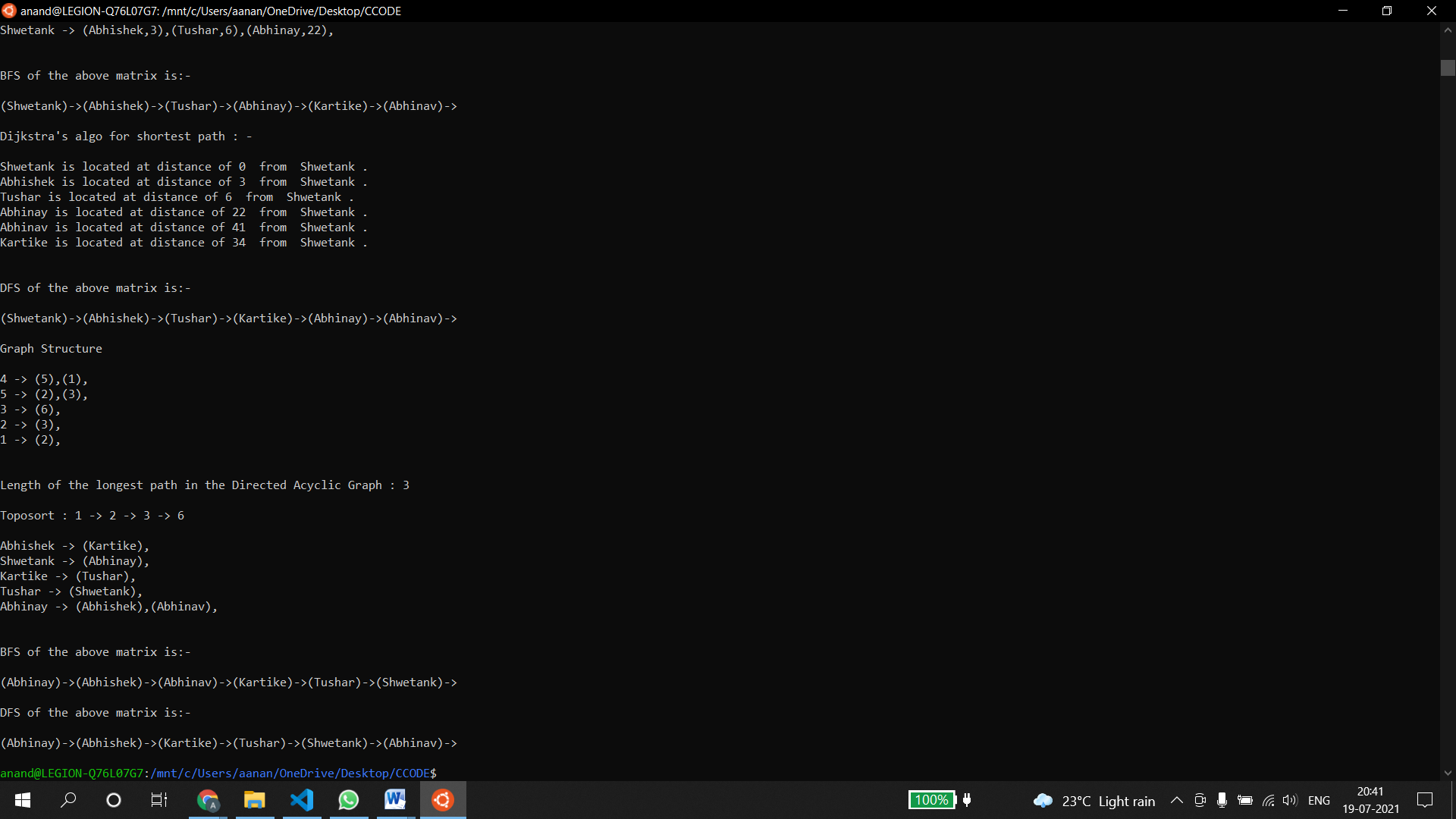
**OUTPUT SCREENSHOTS :**

**Graph G:**



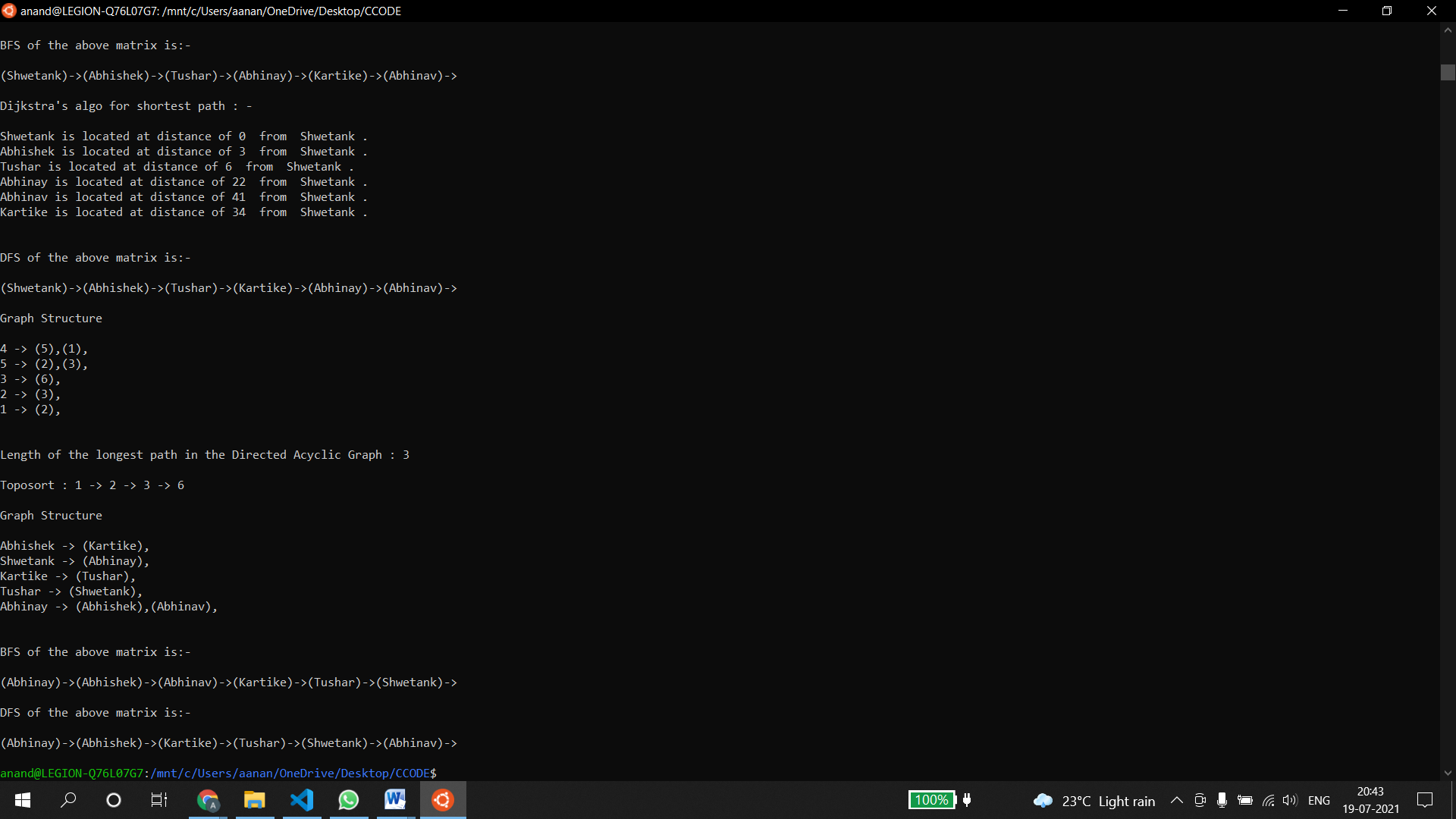
**Here we have created an undirected graph having weights.  
  
Graph has string nodes so we have created an object of type string of the class Graph  
by typing Graph<string> g .  
   
Then we have called the various functions we’ve implemented in our class template .  
  
Firstly we have called the bfs function which traverses the graph and prints the nodes in level order.  
  
Then we have called the dijkstra’s algorithm function which gives us the shortest distance between two nodes.  
  
And then we have called the dfs function which traverses the graph and prints the nodes depth wise.**

**Graph G1:**



**Here we have created a directed graph which is unweighted.  
  
Graph has integer nodes so we have created an object of type int of the class Graph  
by typing Graph<int> g1 .  
   
Then we have called the various functions we’ve implemented in our class template .  
  
Firstly we have called the Longest path in the DAG function which returns us the length of the longest path in the graph created above .  
  
Then we’ve performed topological sort on the above graph by calling the toposort function which pushes the nodes in a stack by traversing them using dfs and hence the nodes are returned in the topological order.**

**Graph G2:**



**Here we have created a directed graph which is unweighted.  
  
Graph has string nodes so we have created an object of type string of the class Graph  
by typing Graph<string> g2 .  
   
Then we have called the various functions we’ve implemented in our class template .  
  
Firstly we have called the bfs function which traverses the graph and prints the nodes in level order.  
  
And then we have called the dfs function which traverses the graph and prints the nodes depth wise.**

**COUNTER MEASURES**

**We have taken a lot of counter measures while making this project and some of those are: -**

** We have made sure that all the algorithms mentioned above can be implemented on both weighted and unweighted graphs.**

** This graph class is implemented using generics**

**so that all the algorithms can be implemented**

**irrespective of the data type of the nodes in the graph.**

**We have taken care of the bidirectional as well as weighted nature of the graph using boolean variables and some case work.**

**SUMMARY**

** We learned a lot during this project, including the implementation of Graphs and the various Algorithms we performed on the same such as Insertion, Dijkstra(shortest path), Cycle Detection , DFS, BFS etc.**

** We also did learn a lot about the concept of OOPS,Generic Programming as we did implement a class and created various data members and member functions in order to carry out the various operations.**

** So, we were able to complete our project**

**GENERICS ON GRAPH.**

** In the end, we were able to create a fully functional project, and able to solve fundamental problems using the concept of graphs.**

**REFERENCES :**

**Books:**

Guide to Competitive Programming

* Antti Laaksonen

# Competitive Programming 3

* Steven and Felix Halim

<https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/>

https://www.geeksforgeeks.org/depth-first-search-or-dfs-for-a-graph/

https://www.geeksforgeeks.org/breadth-first-search-or-bfs-for-a-graph/

https://www.geeksforgeeks.org/topological-sorting/

https://www.geeksforgeeks.org/detect-cycle-in-a-graph/

https://www.geeksforgeeks.org/kruskals-minimum-spanning-tree-algorithm-greedy-algo-2/

[https://www.geeksforgeeks.org/find-longest-path-directed-acyclic-graph/#:~:text=Given%20a%20Weighted%20Directed%20Acyclic,vertices%20in%20the%20given%20graph.&text=The%20idea%20is%20similar%20to,.%2C%20we%20use%20Topological%20Sorting](https://www.geeksforgeeks.org/find-longest-path-directed-acyclic-graph/" \l ":~:text=Given a Weighted Directed Acyclic,vertices in the given graph.&text=The idea is similar to,.%2C we use Topological Sorting).