|  |
| --- |
|  |
| DATA STRUCTURES EC-200 |
| PROJECT TITLE |
|  |
| **UGD NETWORKS GRAPH** |

Table of Contents

[1. Introduction 4](#_Toc43887631)

[2. Description 4](#_Toc43887632)

[2.1. Methodology 4](#_Toc43887633)

[2.2. Results Enter output of your program when executed 4](#_Toc43887634)

[2.3. Limitations 4](#_Toc43887635)

[3. Conclusion 4](#_Toc43887636)

[4. References 4](#_Toc43887637)

# Introduction

**Why the given problem is important?**

* The given problem is very important in order to understand basics of wireless networks, data structures and some algorithms.
* Also shift from actual stations-based node connections to wireless networks where nodes are connected directly without a base station is fast. So, it clarifies new methodologies in this regard.

**Where is it used? State some of its Applications**

* For critical evaluation of performance of network protocols and improvement in their designs, deep understanding of structural properties of wireless networks is essential.
* Many protocols for wireless networks- control, routing, topology, storage/retrieval of information-are based on unit-disk graph (**UGD**)

# Description

## Methodology

**What we already had, additions, modifications:**

* **Lab #13:** Basic implementation and understanding of Graph representation using both matrix and Linked Lists.
* **BFS, DFS, Dijkstra:** pseudo codes and understanding was taken from books cited below altogether mentioned in References and Links. Then, redesigned according to requirements in projects and written in proper c++ syntax.

**Other Learning Resources and for what they are used:**

* **UGD** was understood from research gate via several publications. Link is mentioned below in Websites and Links section References and Links.
* **Syntax errors**, some info about **iterators and STL** built in functionalities was taken from

Geeksforgeeks and stack Overflow mentioned below in References and Links.

* **Image** is taken for understanding purposes from Wikipedia. Link mentioned in

References and Links.

**What we had to do from scratch:**

* **NETWORK Class:**

1. Generation of network by uniformly random plotting nodes on provided area of nxn
2. Shortest path problem implementation
3. It further inherits myGraph\_class for proper functionalities.

**Other Classes explained:**

* **STRUCT (str\_list\_nw):**

1. It contains vertex name, weight, position x and position y.

* **myGraph\_class:**

1. It has vector of lists, where vector stores vertices and list have adjacent nodes of vertex.
2. Vector’s obj is of ADJLIST type and ADJLIST is of str\_list\_nw type.
3. It provides all mentioned functionalities i.e. add Vertex, add Edge, delete Vertex, delete Edge, get Neighbors, getNumberVertices, and getNumberEdges.

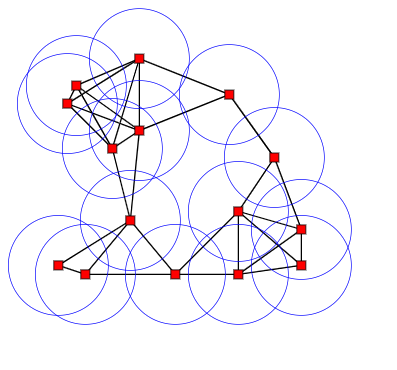
* **ADJLIST class:**

1. It contains variable numlinks for storing number of links.
2. It provides functionalities of increase, decrease in numlinks and size.
3. It inherits STL list class.

* **Main Class:**

1. In main all functionalities are making accessible to user.
2. Switch case is used for accessing functionalities of Graph class and Network class
3. Also, within this Switch case another Switch case is used for further selection of options and choices.

**Pictorial representation of what is being implemented:**



**Figure 2.1:** UDG image

**Execution:**

**A screenshot of a cell phone

Description automatically generated**

## Results Following results are being obtained:

* Generating Graph
* Generating wireless network
* Breadth First Search (BFS)
* Depth First Search (DFS)
* Shortest Path

Output:

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

## Limitations

What didn't work?

# Conclusion

What was accomplished / learned

What you would have done differently

Future work

# References and Links

Include references to all the sources which helped us accomplishing our project.

**BOOKS:**

* Weiss, M.A., 2006. Data Structures and Algorithm Analysis in C++.
* Frank, M.C., 2004. Data Abstraction and Problem Solving with C++. US.
* Dasgupta, S., Papadimitriou, C.H. and Vazirani, U.V., 2008. *Algorithms* (p. 173). New York: McGraw-Hill Higher Education.

**LINKS and WEBSITES**:

* [**https://en.wikipedia.org/wiki/Unit\_disk\_graph**](https://en.wikipedia.org/wiki/Unit_disk_graph)
* [**https://www.geeksforgeeks.org/**](https://www.geeksforgeeks.org/)
* [**https://www.researchgate.net/**](https://www.researchgate.net/)
* [**https://stackoverflow.com/**](https://stackoverflow.com/)

**FOR CITATION:**

* [**https://scholar.google.com/**](https://scholar.google.com/)

**HELP REGARDING ADDING LINKS IN WORD:**

* [**https://support.microsoft.com/en-us/office/create-or-edit-a-hyperlink-5d8c0804-f998-4143-86b1-1199735e07bf**](https://support.microsoft.com/en-us/office/create-or-edit-a-hyperlink-5d8c0804-f998-4143-86b1-1199735e07bf)