CS2023 - Inclass Lab

Week 10 – Graphs

Submission by: Sajeev Kugarajah (210554M)

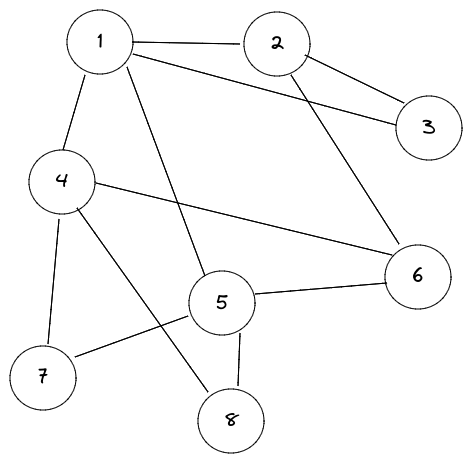


Figure 1: Graph for Section 1

Section 1 : Implementing Graph ADT

1. Write the adajaceny list representation for the graph in Fig1

1 => [2, 3, 4, 5]

2 => [1, 3, 6]

3 => [1, 2]

4 => [1, 6, 7, 8]

5 => [1, 6, 7, 8]

6 => [2, 4, 5]

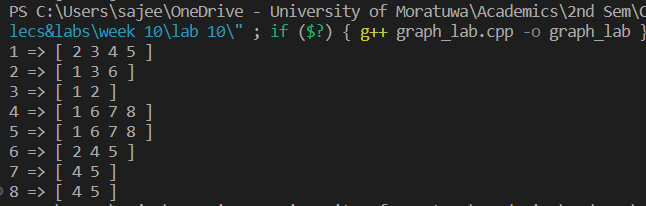
7 => [4, 5]

8 => [4, 5]

1. By using comments provided in the code, complete the following *Node (struct), addedge, print functions*

Source code has been uploaded to GitHub.

1. Create graph object and add the graph in Fig.1.
2. Print the adjacency list using the *print* function you implemented and take screenshot.



1. What is the change you will make in the *addedge* function so that Graph ADT could accept directed graphs.(Instead of accepting undirected graph, we need to accept directed graph). Write addedge altered function as your answer below.

void addedge(int u, int v, **int directed = 0**)

// 0 = undirected // 1 = directed

    {

if(directed){

// select node u and push v into u's neighbour

nodes[u - 1].neighbours.push\_back(v);

} else {

// select node u and push v into u's neighbour

// select node v and push u into v's neighbour

nodes[u - 1].neighbours.push\_back(v);

nodes[v - 1].neighbours.push\_back(u);

}

    }

Section 2 : Working out link prediction, no coding required

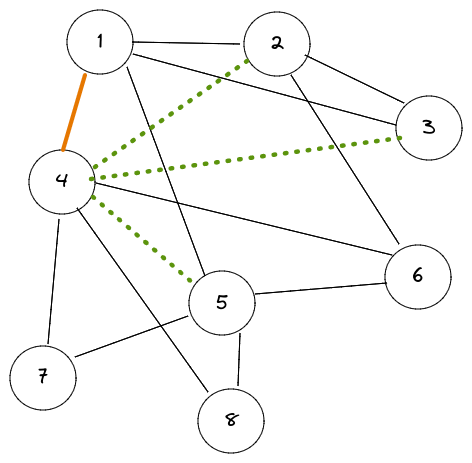


Figure 2: Graph for Section 2

To predict whether two nodes may have edge between them in the future, we must define a similarity score between the 2 nodes. Use the following similarity equation to calculate the similarity between 2 nodes,

# of shared neighbours between a,b *a* ∩ *b*

*Sim*(*a,b*) = =

Total neighbours in a,b *a* ∪ *b*

Refer graph in Fig.2 to answer the question below.

Lets assume graph in Fig.2 is a social network graph of a social media platform, where nodes denote people and edges between them indicate that they are connected as friends. Node 1 and Node 4 just became friends, which of the neighbours of Node 1 will you suggest for Node 4 (in other word predict which neighbour of Node 1 can have a edge with Node 4). Utilize the similarity function provided to justify the answer.

**Answer:**

Let’s take a look at Nodes 4, 2, 3, 5 and its’ neighbours.

4 => [1, 6, 7, 8]

2 => [1, 3, 6]

3 => [1, 2]

5 => [1, 6, 7, 8]

now apply the similarity function to the nodes.

Sim(4,2) = 2/5

Sim(4,3) = 1/5

Sim(4/5) = 4/4

Hence from this similarity function we can say that nodes 4 and 5 are having a high similarity score between them. So, node 4 and 5 have a higher probability to have an edge between them in the near future.