CS2023 - Inclass Lab

Week 10 - Graphs

Note: You are required to answer the below questions and submit a PDF to the submission link provided under this week before the deadline (no extensions will be provided). You can either write / type your answers, but either way your answers should be readable.

Add the link to the GitHub repository

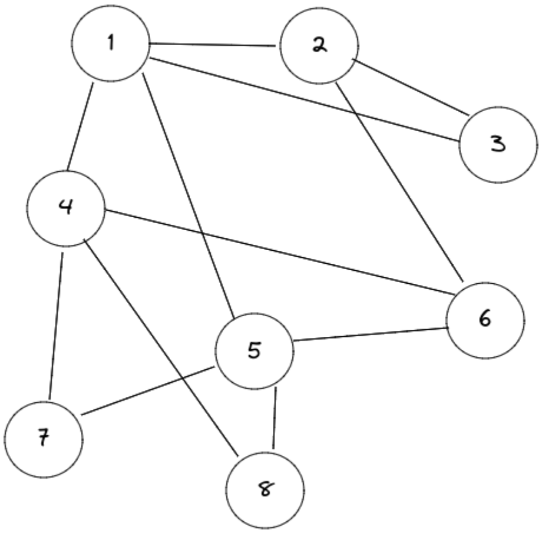


Figure 1: Graph for Section 1

Lab instruction

Please download the lab materials from the lab 10 section We will be implementing a Graph Abstract Data Type (ADT) in section 1 and section 2 we will be working a problem of link prediction. Section 1 requires coding, Section 2 DO NOT require coding.

Section 1 : Implementing Graph ADT

Expected submission

1. Write the adajaceny list representation for the graph in Fig1
2. By using comments provided in the code, complete the following *Node (struct), intializenodes, addedge, print functions*
3. Create graph object and add the graph in Fig.1.
4. Print the adjacency list using the *print* function you implemented and take screenshot.
5. 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.

Section 2 : Working out link prediction, no coding required

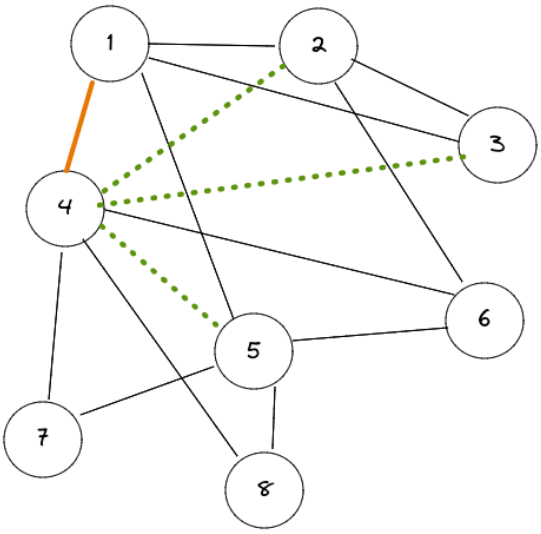


Figure 2: Graph for Section 2

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,

$$Sim(a,b) = \frac{\text{\# of shared neighbours between a,b}}{\text{Total neighbours in a,b}} = \frac{a \cap b}{a \cup b}$$

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.

Answers

Section 1:

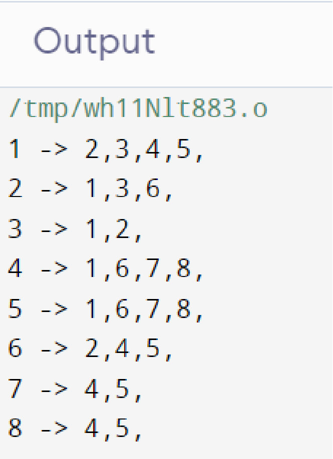
1)

    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]

2)

3)

4)



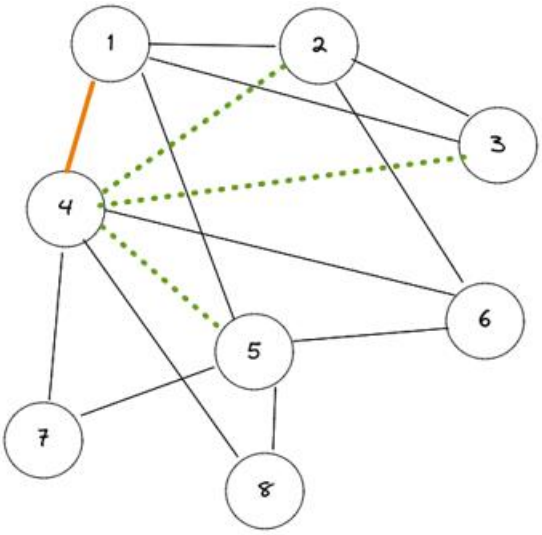
Output

```
/tmp/wh11Nlt883.o
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,
```

5)

```
void addedge(int u, int v){
    //select node u and push v into u's neighbour
    nodes[u].neighbours.push_back(v);
}
```

Section 2:



- Neighbors of 1 -> 2,3,5,(4)
- Let’s calculate similarities between 4 and neighbors of 1.

  Sim(4,2) = $\frac{2}{5}$

  Sim(4,3) = $\frac{1}{5}$

  Sim(4,5) = $\frac{4}{4}$
- The highest similarity is between 4 and 5. Hence 5 is the best suggestion.