

# In-class exercise (In20-S4-CS2023)

## Lab 10

ID – 200709K

Wickramanayake R.S.D.

Q1:

Node 1 : -> 2 -> 3 -> 4 -> 5

Node 2 : -> 1 -> 3 -> 6

Node 3 : -> 1 -> 2

Node 4 : -> 1 -> 6 -> 7 -> 8

Node 5 : -> 1 -> 6 -> 7 -> 8

Node 6 : -> 2 -> 4 -> 5

Node 7 : -> 4 -> 5

Node 8 : -> 4 -> 5

Q4:

```
PS D:\University\ENTC - sem 4\Data Structures and Algorithms\labs\lab10> cd "d
g++ lab_10.cpp -o lab_10 } ; if ($?) { .\lab_10 }
Node1 neighbours:  -> 2 -> 3 -> 4 -> 5
Node2 neighbours:  -> 1 -> 3 -> 6
Node3 neighbours:  -> 1 -> 2
Node4 neighbours:  -> 1 -> 6 -> 8 -> 7
Node5 neighbours:  -> 1 -> 6 -> 7 -> 8
Node6 neighbours:  -> 2 -> 4 -> 5
Node7 neighbours:  -> 4 -> 5
Node8 neighbours:  -> 4 -> 5
PS D:\University\ENTC - sem 4\Data Structures and Algorithms\labs\lab10> █
```

Q5:

```
void addedge(int u, int v) {
nodes[u].neighbours.push_back(v);
}
```

## Section 2

- Node 1 neighbors:  
2,3,4,5
- Node 2 neighbors:  
1, 6, 7, 8

Sim(4, 2):

- Number of shared neighbors = 0
- Total number of neighbors = 7 (Node 4 - 4 neighbors, Node 2 - 3 neighbors)

$$\text{Sim}(4, 2) = 0 / 7 = 0$$

Sim(4, 3):

- Number of shared neighbors = 0
- Total number of neighbors = 6 (Node 4 - 4 neighbors, Node 3 - 2 neighbors)

$$\text{Sim}(4, 3) = 0 / 6 = 0$$

Sim(4, 5):

- Number of shared neighbors = 2 (Shared Node 6, Node 7)
- Total number of neighbors = 7 (Node 4 - 4 neighbors, Node 5 - 4 neighbors)

$$\text{Sim}(4, 5) = 2 / 7 \approx 0.285714$$

**Therefore, Node 5 is a potential candidate to have an edge with node 4.**