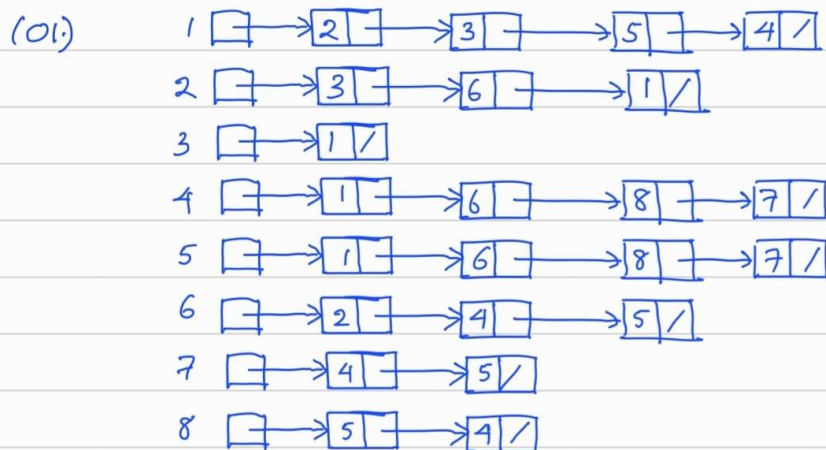


InClass Lab 10

200356A

Section 1

Section 01



04.)

```
PS C:\Users\Dulan Lokugeegana> cd "c:\Users\Dulan Lokugeegana\OneDrive - University of Moratuwa\Sem 4 Dulan\CS2023_data_structures_and_algorithms\InClass\LAB10 May16\" ; if ($?) { g++ graph_lab.cpp -o graph_lab } ; if ($?) { .\graph_lab }
1 : 2, 3, 5, 4,
2 : 1, 3, 6,
3 : 1, 2,
4 : 1, 6, 8, 7,
5 : 1, 7, 8, 6,
6 : 2, 4, 5,
7 : 4, 5,
8 : 4, 5,
PS C:\Users\Dulan Lokugeegana\OneDrive - University of Moratuwa\Sem 4 Dulan\CS2023_data_structures_and_algorithms\InClass\LAB10 May16>
```

05.)

```
void addedge(int u, int v){
    //select node u and push v into u's neighbour
    nodes[u-1].neighbour.push_back(v); // direction is considered
    //select node v and push u into v's neighbour
    // nodes[v-1].neighbour.push_back(u);
}
```

Section B

When we considered node 1 and 4, there are 4 neighbours for node 1. Then we can use similarity function for node 4 and each neighbours of node 1 to find which one is the most suitable suggestion.

a\b	2	3	5	4
4	$2/5=0.4$	$1/5=0.2$	$4/4=1$	Not considered

From the above similarity graph we can see that node 5 is the best suggestion for node 4.

Github

<https://github.com/Dulan24/S4-CS2023-DSA-labs>