Lab Report – Week 10

CS2023 Data Structures and Algorithms

Dept. of Computer Science and Engineering, University of Moratuwa

Name: Tharindu Perera Index Number: 210472J

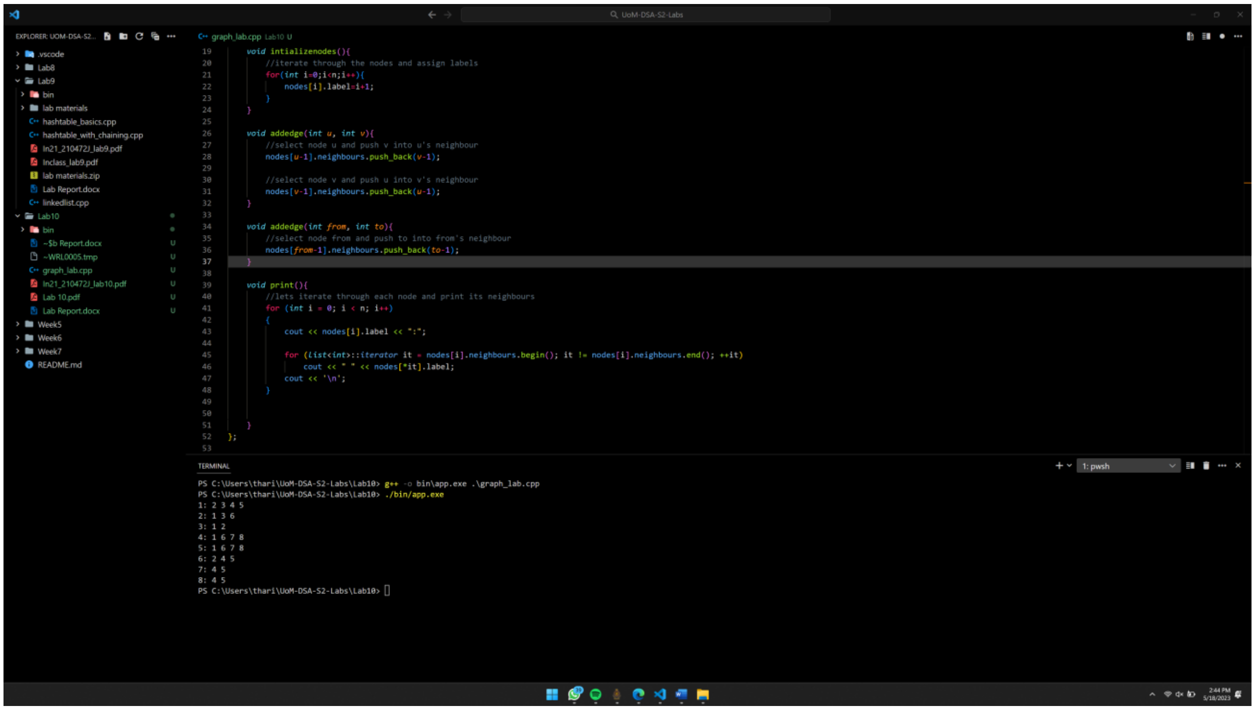
Section1: Implementing Graph ADT

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

4.

```
PS C:\Users\thari\UoM-DSA-S2-Labs\Lab10> g++ -o bin\app.exe .\graph_lab.cpp
PS C:\Users\thari\UoM-DSA-S2-Labs\Lab10> ./bin/app.exe
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
PS C:\Users\thari\UoM-DSA-S2-Labs\Lab10>
```



5.

```cpp
void addedge(int from, int to){
    //select node from and push to into from's neighbour
    nodes[from-1].neighbours.push_back(to-1);
}
```

*Index is decremented to fit the zero based indexing in the list.

Section2: Working out link prediction, no coding required

Traverse through each of the neighbours of node 1 and and for each of them travers through their neighbours and save it to a set data structure. Next traverse through the neighbours of node 4 and save it to a nother set data structure. Save bothe of those to another set structure to get all naighbours of both and get the count of it (n_all) get the counts of the previous two sets n_1 nd n_4 we can get the count of neighbours that both 1 and 4 have in common by n_com = n_1 + n_4 - n_all then find similarity of each of 1s neighbours and 4 by similarity = n_com / n_all. Suggest 1s neigh bour with the highest similarity with 4.

GitHub Link : [Tharindu6516/UoM-DSA-S2-Labs (github.com)]()