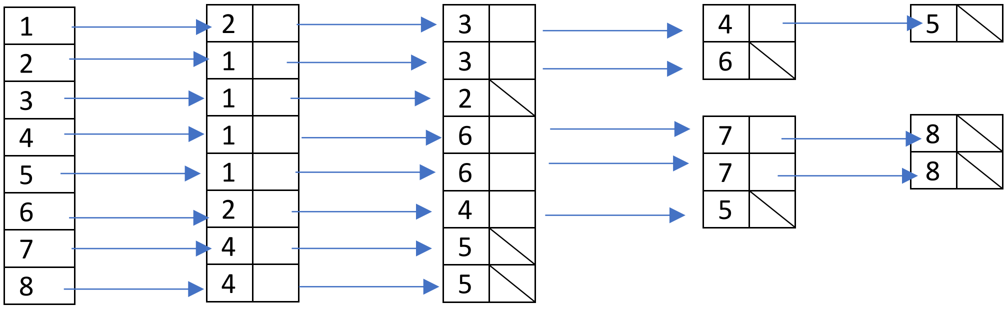
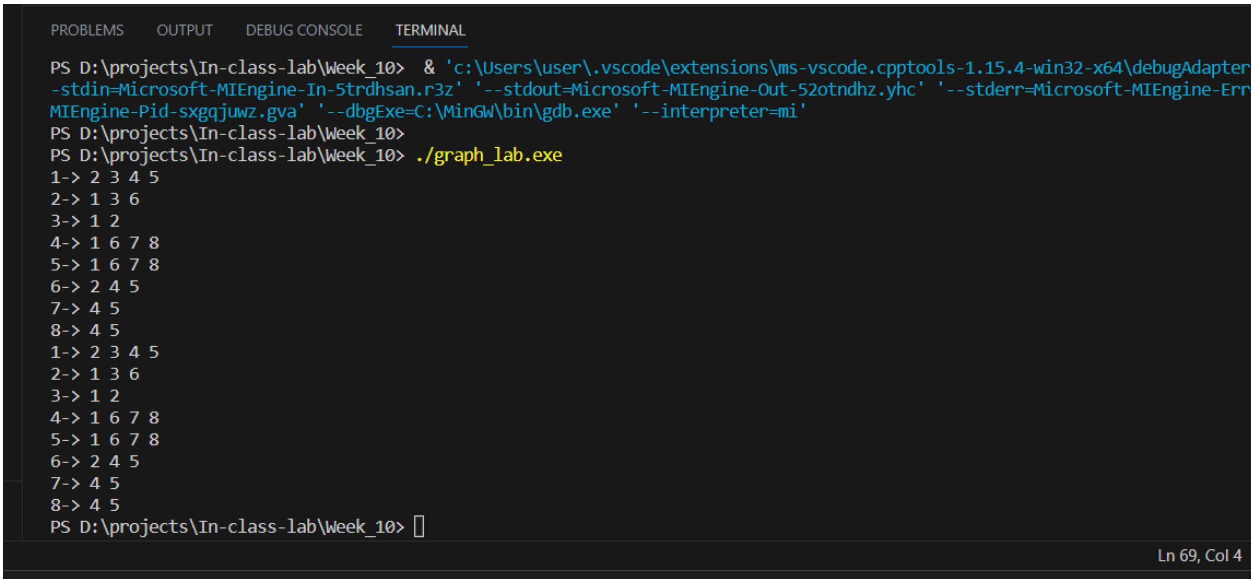
In class Lab – Data Structures and Algorithms

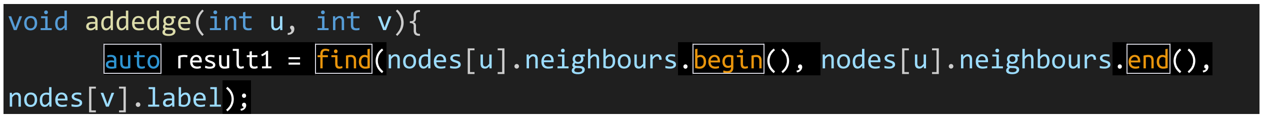
Index no- 210670N

Section 1





```
PS D:\projects\In-class-lab\Week_10>  & 'c:\Users\user\.vscode\extensions\ms-vscode.cpptools-1.15.4-win32-x64\debugAdapter
-stdin=Microsoft-MIEngine-In-5trdhsan.r3z' '--stdout=Microsoft-MIEngine-Out-52otndhz.yhc' '--stderr=Microsoft-MIEngine-Err
MIEngine-Pid-sxgqjuwz.gva' '--dbgExe=C:\MinGW\bin\gdb.exe' '--interpreter=mi'
PS D:\projects\In-class-lab\Week_10>
PS D:\projects\In-class-lab\Week_10> ./graph_lab.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
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 D:\projects\In-class-lab\Week_10> 
```



```
void addedge(int u, int v){
        auto result1 = find(nodes[u].neighbours.begin(), nodes[u].neighbours.end(),
nodes[v].label);
```

```
       auto result2 = find(nodes[v].neighbours.begin(), nodes[v].neighbours.end(),
nodes[u].label);

       if( result1 != nodes[u].neighbours.end()){
         nodes[u].neighbours.push_back(nodes[v].label);
       }
       If(result2 != nodes[v].neighbours.end()) {
         nodes[v].neighbours.push_back(nodes[u].label);
       }
         //select node u and push v into u's neighbour

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

Section 2

Sim(4,2) = 2/ 5

Sim(4, 3) = 1/5

Sim(4, 5) = 4/8 = ½

Out of all neighbours of node 4, node 5 has the highest chance of having an edge with node 4.