THE UNIVERSITY
OF LAHORE
ISLAMABAD
CAMPUS

# Data Structures and Algorithms
# ( CS09203 )

## Lab Report

| | |
|---|---|
| Name: | Jassim Bashir |
| Registration #: | SEU-F16-101 |
| Lab Report #: | 08 |
| Dated: | 28-05-2018 |
| Submitted To: | Sir. Usman Ahmed |

The University of Lahore, Islamabad Campus
Department of Computer Science & Information Technology

# Experiment # 8
# Implementation of directed,undirected and weighted graph

**Objective**
The objective of this session is to create the graphs directed,undirected and weighted.

**Software Tool**
1. I use Code Blocks with GCC compiler.

## Theory

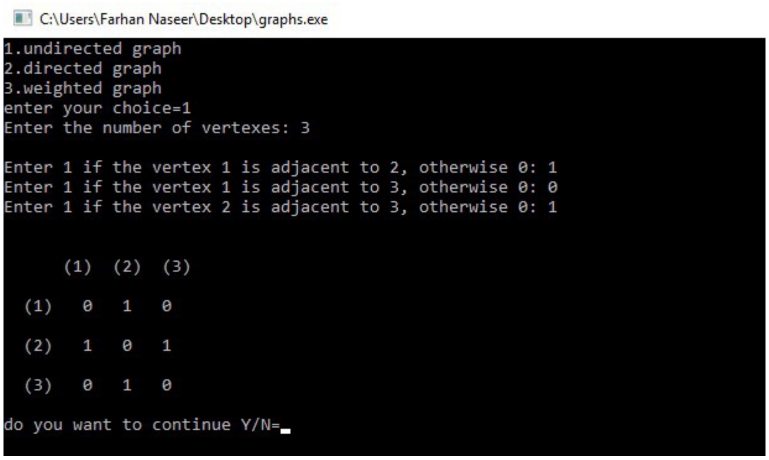
This section discusses how to create the graph and tell the number of edges and vertices . Graphs are used to model electrical circuits, chemical compounds, highway maps, and so on. They are also used in the analysis of electrical circuits, finding the shortest route, project planning, linguistics, genetics, social science, and so forth Undirected Edge - An undirected egde is a bidirectional edge. If there is a undirected edge between vertices A and B then edge (A , B) is equal to edge (B , A). Directed Edge - A directed egde is a unidirectional edge. If there is a directed edge between vertices A and B then edge (A , B) is not equal to edge (B , A). Weighted Edge - A weighted egde is an edge with cost on it.

## Task

### Procedure: Task 5

Write a C++ code using functions for the following operations. 1.Creating directed graph. 2.Creating undirected graph. 3.Creating weighted graph

## 2.2



```
C:\Users\Farhan Naseer\Desktop\graphs.exe
1.undirected graph
2.directed graph
3.weighted graph
enter your choice=1
Enter the number of vertexes: 3

Enter 1 if the vertex 1 is adjacent to 2, otherwise 0: 1
Enter 1 if the vertex 1 is adjacent to 3, otherwise 0: 0
Enter 1 if the vertex 2 is adjacent to 3, otherwise 0: 1


     (1)  (2)  (3)

  (1)   0   1   0

  (2)   1   0   1

  (3)   0   1   0

do you want to continue Y/N=
```

Figure 1: output

```
include<iostream> #include<iomanip>

using namespace std ;
  void PrintMat( int mat [ ] [ 2 0 ] , int n, int weight [ ] [ 2 0 ] )
{
                int i ,      j ;

              cout<<”\n\n”<<setw(4)<<”” ; for ( i =
              0; i < n; i++)
                            cout<<setw(3)<<”(”<<i+1<<”)” ;
              cout<<”\n\n” ;

                for ( i = 0;        i < n;       i++)
              {
                            cout<<setw(3)<<”(”<<i+1<<”)” ; for ( j =
                            0; j < n; j++)
                            {
                                                   cout<<setw(4)<<mat[ i ] [ j]<<setw(3)<<”  weight              is=”
                            }
                            cout<<”\n\n” ;
```

```
            }
}
void PrintMat( int mat [ ] [ 2 0 ] , int n)
{
              int i ,      j ;

            cout<<”\n\n”<<setw(4)<<”” ; for ( i =
            0; i < n; i++)
                        cout<<setw(3)<<”(”<<i+1<<”)” ;
            cout<<”\n\n” ;

              for ( i = 0;        i < n;      i++)
            { cout<<setw(3)<<”(”<<i+1<<”)” ; for ( j = 0; j <
                        n; j++)
                        { cout<<setw(4)<<mat[ i ] [ j ] ;
                        } cout<<”\n\n” ;
            }
}

int main()
{
            char ch ; int choice ; int      mat [20][20] ,
                weight [ 2 0 ] [ 2 0 ] ;
            do{
            cout<<”1. undirected graph”<<endl ;
            cout<<”2. directed graph”<<endl ; cout<<”3.
            weighted graph”<<endl ; cout<<”enter your
            choice=” ; cin>>choice ; switch( choice ){ case
            1:{ int i ,    j , v ;

            cout<<”Enter the number of           vertexes :” ; cin>>v

            ; int           mat [20][20] ,   weight [ 2 0 ] [ 2 0 ] ;

            cout<<”\n” ;

              for ( i = 0;       i < v ;      i++)
            {
                          for ( j = i ;       j < v ;      j++)
```

```
                                             cin>>mat[ i ] [ j ] ; mat[ j ] [ i ] = mat[ i ] [

                                             j ] ;


                      } else                 mat[ i ] [ j ] = 0;

           }
}

PrintMat(mat, v );} break;
case 2:
           {
                                    int i ,    j , v ;

cout<<"Enter the number                    of      vertexes : " ;
c

i

n

>

>

v



;



i

n

t
```

m

a

t

[

2

0

]

[

2

0

]

```
;

cout<<”
\n” ;
for ( i =
0;  i < v ;
       i++)
{
 for ( j =    j++)
0;  j < v ;
             {
  cout<       1              i f             the            vertex        ”<<i+1      is
 <”Ente                                                                   <<”
          r
                                { if ( i != j ){ cout<<”Enter 1    i f        the vertex”<<i+1<<” is
                                                                    cin>>mat[ i ] [ j ] ;
                                }
                  }

                  PrintMat(mat, v );}
                                                   break;
                  case 3:
                                {                                   {
                                                       int i ,      j , v ;

                  cout<<”Enter the number of             vertexes :” ; cin>>v
                  ; int            mat [ 2 0 ] [ 2 0 ] ;

                  cout<<”\n” ;
for ( j     j < v ;    j++)
= 0;
{
                        cout         i  t     ve     ”<<i
                        <<”             h     rte    +1<<
                        Enter        f  e     x      ”
cin
>>
mat
[ i ]
[ j ]
;
if
(m
at[
i ]
[ j
]
==
1){
 c      the    weight=” ;
 o
 u
 t
<
<
”
 e
 n
 t
 e
 r
                 for ( i = 0;       i < v ;      i++)
               {




                                                                   cin>>weight [ i ] [ j ] ;
                                                         } else
                                                           weight [ i ] [ j ]=0;
                         }
               }

               PrintMat(mat, v , weight );} break;
                                   default :
                                                          cout<<” invalid ”<<endl ;
```

```
                              }
                              }
                              cout<<”do you want to continueY/N=” ; cin>>ch ;
}while (( ch==’Y’ ) | | ( ch==’y ’ ));
}
```

## Conclusion

In today lab we have discussed how we can create a Graphs directed ,undirected and weighted graph and how to display it on a screen by a code.