

# DATA STRUCTURE AND ALOGRITHUM

**Lab Report**

Name: Ahsan Khurshid

Registration #: SEU-F16-173

Lab Report #: 08

Dated: 5-28-2018

Submitted To: Mr. Usman Ahmed

The University of Lahore, Islamabad Campus

Department of Computer Science & Information Technology

**Experiment # 1**

**GRAPH**

**Objective**

To understand the implementation of adjancy matrix.

**Software Tool**

1.

DEV C++

# Theory

There are 3 function creted which perfom differents tasks 1. DIRECTED

GRAPH

1. UNDIRECTED GRAPH
2. WEIGHTED GRAPH

# Task

## Procedure: Task 1

*#include<iostream> #include<iomanip>* using namespace std ;

void PrintMat( **int** mat [ ] [ 2 0 ] , **int** n, **int** weight [20][20])

{

**int** i , j ;

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

cout*<<*setw(3)*<<*”(”*<<*i+1*<<*”) ” ;

cout*<<*”\n\n” ;

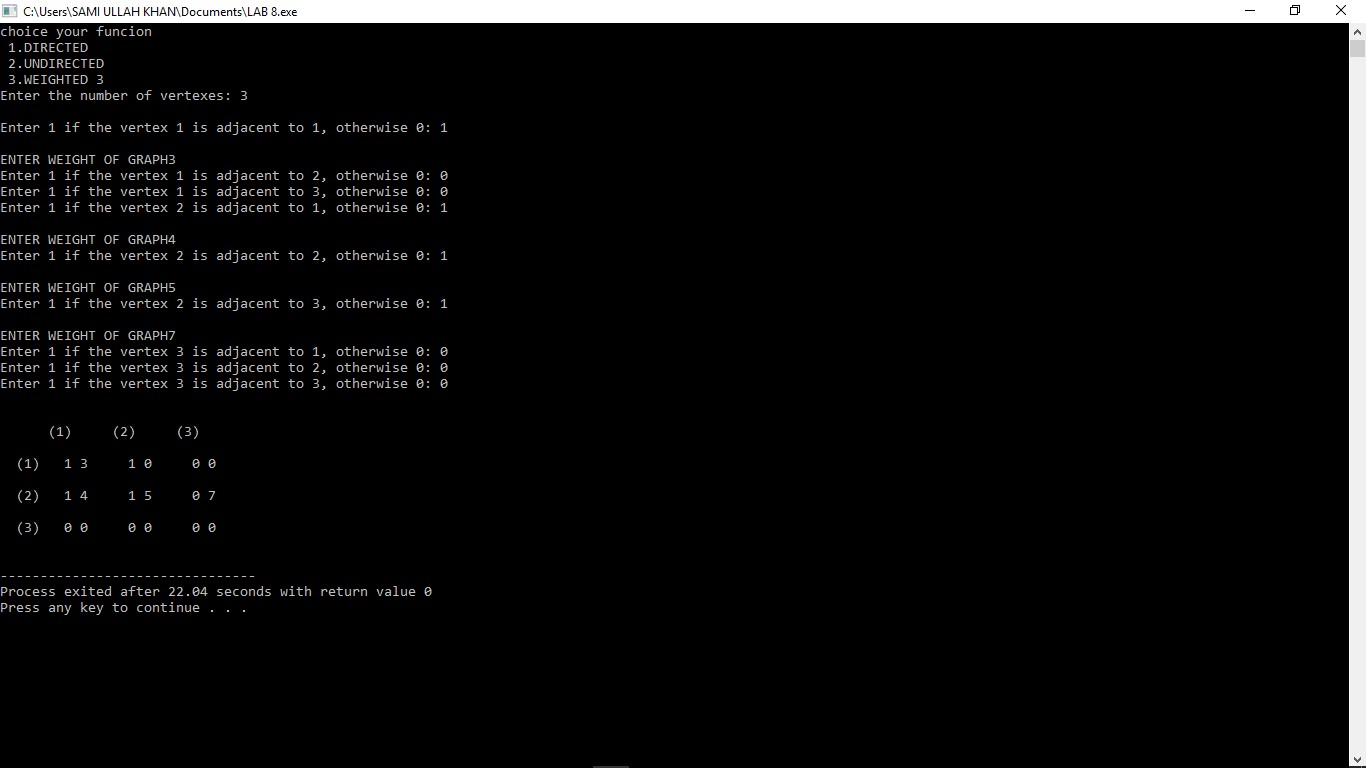


Figure 1: Time Independent Feature Set

// Print 1 **if** the corresponding vertexes are connected otherwise 0 **for** ( i = 0; i *<* n; i++)

{

cout*<<*setw(3)*<<*”(”*<<*i+1*<<*”)” ; **for** ( j = 0; j *<* n; j++)

{

cout*<<*setw(4)*<<*mat[ i ] [ j]*<<*”” *<<*weight [ i ] [ j]*<<*” ” ;

}

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” ;

// Print 1 **if** the corresponding vertexes are connected otherwise 0 **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()

{

**int** n; **int** i , j , v ; **int** mat [ 2 0 ] [ 2 0 ] ; **int** weight [ 2 0 ] [ 2 0 ] ;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| cin*>>*n; switch (n) { |  |  |  |  |  |
| case | 1: |  |  |  |  |
| cout*<<*”Enter | the | number | of | vertexes : | ” ; |

cout*<<*”choice your funcion \n 1.DIRECTED \n 2.UNDIRECTED \n 3.WEIGH

a

|  |  |  |  |
| --- | --- | --- | --- |
| cin*>>*v ; **int** mat [ 2 0 ] [ 2 0 ] ; cout*<<*”\n” ; |  |  |  |
| // Take **input** of the adjacency **for** ( i = 0; i *<* v ; i++)  { | of each pair | of | vertexes . |
| **for** ( j = 0; j *<* v ; j++)  { | |

{

cout*<<*”Enter 1 i f the vertex”*<<*i+1*<<*” is cin*>>*mat[ i ] [ j ] ;

// mat[ j ] [ i ] = mat[ i ] [ j ] ;

}

}

}

PrintMat(mat, v ); **break**; case 2:

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

cout*<<*”\n” ;

// Take **input** of the adjacency of each pair of vertexes .

**for** ( i = 0; i *<* v ; i++)

{

**for** ( j = i ; j *<* v ; j++)

{

**if** ( i != j )

a

{

cout*<<*”Enter 1 i f the vertex”*<<*i+1*<<*” is cin*>>*mat[ i ] [ j ] ; mat[ j ] [ i ] = mat[ i ] [ j ] ;

} **else**

mat[ i ] [ j ] = 0;

}

}

PrintMat(mat, v ); **break**;

case 3: **int** i , j , v ;

cout*<<*”Enter the number of vertexes :” ;

cin*>>*v ; cout*<<*”\n” ;

// Take **input** of the adjacency of each pair of vertexes .

**for** ( i = 0; i *<* v ; i++)

{

|  |  |  |
| --- | --- | --- |
| **for** ( j = 0;  {  { | j *<* v ; | j++) |

cout*<<*”Enter 1 i f the vertex”*<<*i+1*<<*” is cin*>>*mat[ i ] [ j ] ;

mat[ j ] [ i ] = mat[ i ] [ j ] ;

**if** (mat[ i ] [ j]==1)

{

a

cout*<<*endl*<<*”ENTER WEIGHTOF GRAPH” ; cin*>>*weight [ i ] [ j ] ;

} **else** weight [ i ] [ j ]=0;

}

}

}

PrintMat(mat, v , weight ); **break**;

default :

cout*<<*”INVALID: ” ;

}

**return** 0;

}

# Conclusion

in this lab we perform 3 differents task of graph and well understand them in the lab.