Mini project

Group : 07 (Mini Project 1)

CSE 106

Discrete Mathematics

Section : 01

Submitted to:

Md. Mozammel Huq Azad Khan, PhD

Professor, CSE Department, East West University

Submitted by :

Proyas Pratim Nandi Samya

ID: 2024-3-60-491

Md. Shakib Hassan

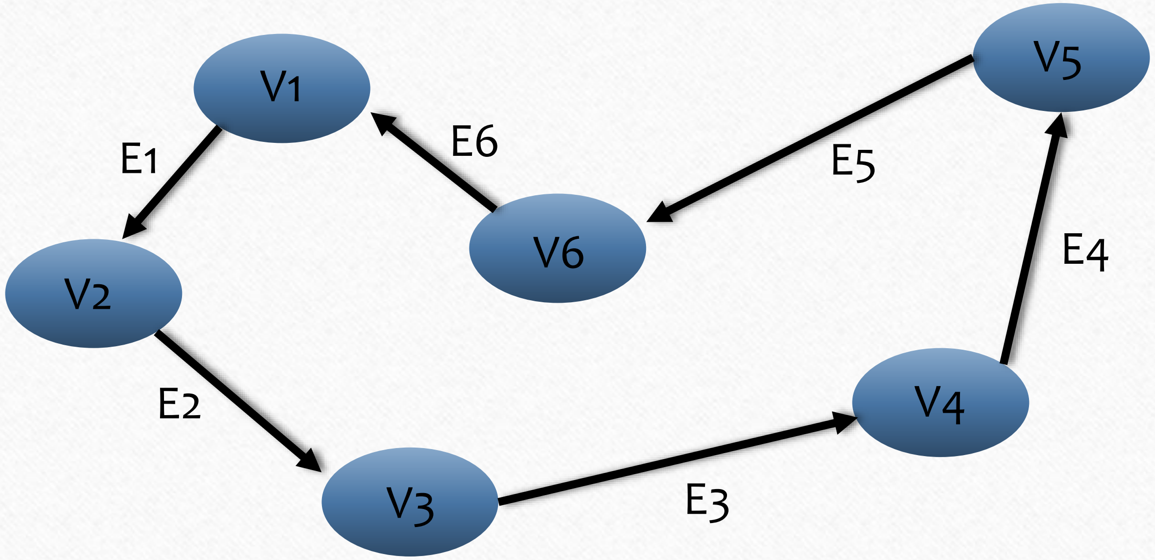
ID: 2024-3-60-676

Tahsin Mahmud

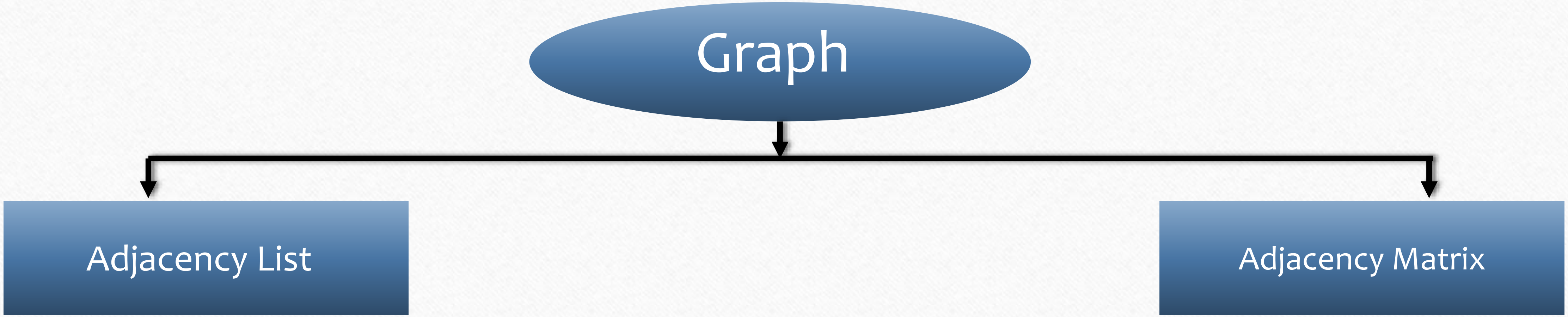
2024-3-60-698

Directed Graph :

A directed graph is defined by a set of vertices and a set of directed edges.



Presenting Graph



<u>Adjacency List</u> : A adjacency list is a way of representing a graph using a collection of lists or array.

<u>Adjacency Matrix</u> : An adjacency matrix is a way of representing a graph using a 2D array.

```c
#include <stdio.h>
#include <stdlib.h>
#include<time.h>
int matrix[8000][8000];row,colum,n;
int main()
{
    double start,end,time;
    start=clock();
    printf("Enter the amount of vertices= ");
    scanf("%d",&n);
    fill();
    result();
    end=clock();
    time=( end- start);
    printf("\nTime taken = %.2f sec.",time);
}
void fill()
{
    srand(time(NULL));
    for(row=1;row<=n;row++)
        {
            for(colum=1;colum<=n;colum++)
        matrix[row][colum]=rand()%2;
        }
}
void result()
{
    int in_degree=0,out_degree=0;
    for(row=1;row<=n;row++)
    {
        for(colum=1;colum<=n;colum++)
            in_degree=matrix[row][colum]+in_degree;
    }
    for(row=1;row<=n;row++)
    {
        for(colum=1;colum<=n;colum++)
            out_degree=matrix[row][colum]+out_degree;
    }
    printf("Total In Degree= %d \nTotal Out Degree= %d",in_degree,out_degree);
}
```

Outputs :

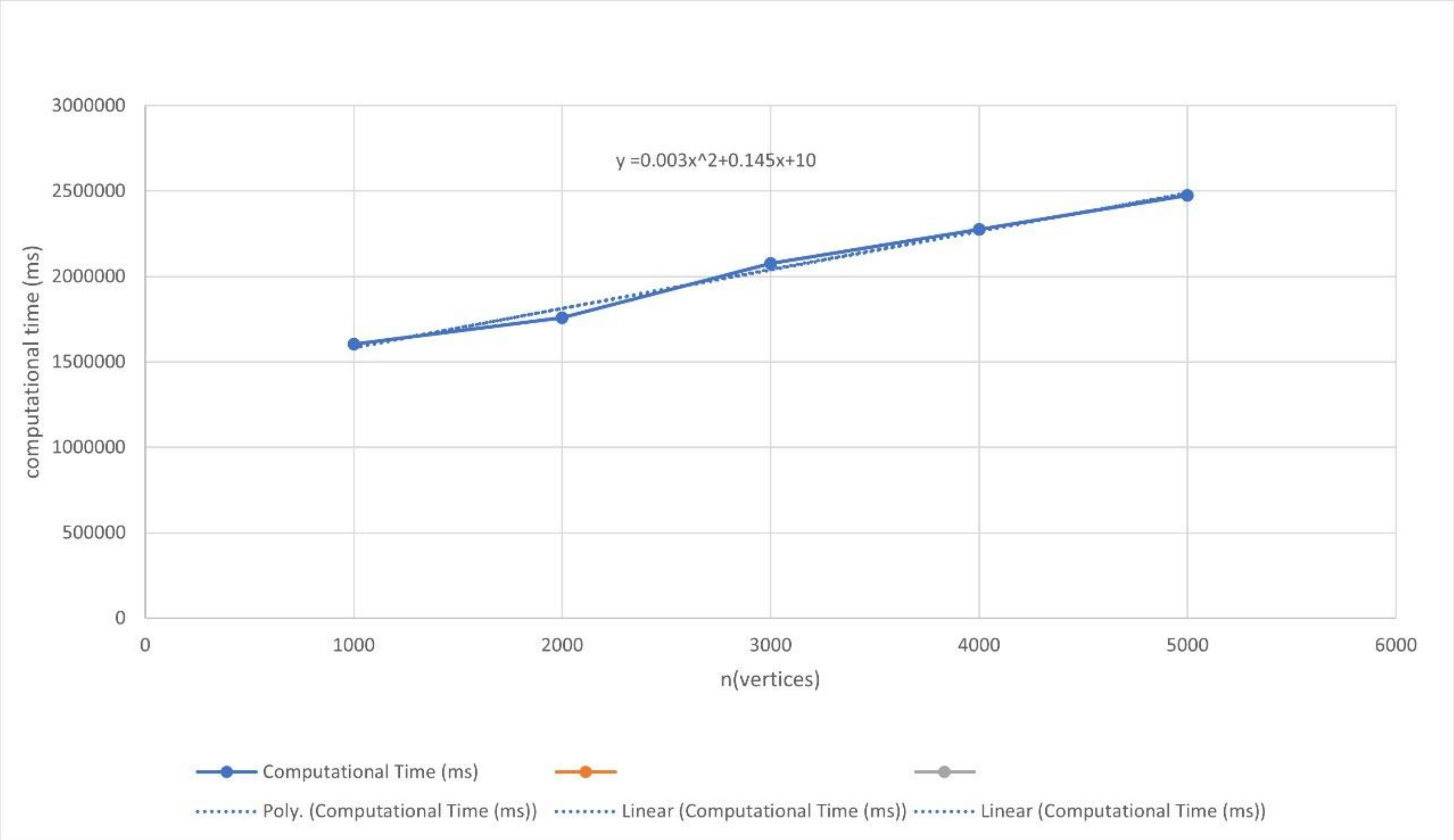
```
"D:\CSE 106\Mini project\bin\
Enter the amount of vertices= 1000
Total In Degree= 499992
Total Out Degree= 499992
Time taken = 1603.00 sec.
Process returned 0 (0x0)   execution time : 1.690 s
Press any key to continue.
```

```
"D:\CSE 106\Mini project\bin\
Enter the amount of vertices= 2000
Total In Degree= 1999943
Total Out Degree= 1999943
Time taken = 1758.00 sec.
Process returned 0 (0x0)   execution time : 1.772 s
Press any key to continue.
```

```
"D:\CSE 106\Mini project\bin\
Enter the amount of vertices= 3000
Total In Degree= 4500130
Total Out Degree= 4500130
Time taken = 2077.00 sec.
Process returned 0 (0x0)   execution time : 2.094 s
Press any key to continue.
```

```
"D:\CSE 106\Mini project\bin\
Enter the amount of vertices= 4000
Total In Degree= 8000021
Total Out Degree= 8000021
Time taken = 2275.00 sec.
Process returned 0 (0x0)   execution time : 2.300 s
Press any key to continue.
```

```
"D:\CSE 106\Mini project\bin\
Enter the amount of vertices= 5000
Total In Degree= 12500100
Total Out Degree= 12500100
Time taken = 2475.00 sec.
Process returned 0 (0x0)   execution time : 2.502 s
Press any key to continue.
```



y =0.003x^2+0.145x+10
computational time (ms)
3000000
2500000
2000000
1500000
1000000
500000
0
0
1000
2000
3000
4000
5000
6000
n(vertices)
Computational Time (ms)
Poly. (Computational Time (ms))
Linear (Computational Time (ms))
Linear (Computational Time (ms))

Theoretical Time Complexity :

Time complexity represents how the computational time of a program increases as the input size grows.

We use an **n × n** adjacency matrix to represent the graph.

To generate the graph, we fill an **n × n** matrix, which requires **O(n^2)** operations.

For each of the n vertices, we perform **O(n^2)** operations to calculate both in-degree and out-degree.

Total operations = O(n) × **O(n)**
= O(n^2)

Total time complexity = O(n^2) + O(n^2)
= O(n^2)

Experimental time complexity (from graph) :

The equation of the trendline was of the form :

$$Y = 0.003x^2 + 0.14x + 10$$

The highest degree term in the equation is **x^2**, which confirms that the time complexity grows quadratically with n.

The trendline shows a quadratic relationship between the computational time and n.

The experimental results also suggest that the complexity of the program is **$O(n^2)$**.

Comparison : Theoretical vs Experimental :

From analyzing the code, we determined that the time complexity is **$O(n^2)$**.

From the excel graph and the polynomial trendline, confirming that the experimental time complexity ia also **$O(n^2)$**.

Both the theoretical and experimental time complexity of the program are **$O(n^2)$**.

Thank You