Course Name: Discrete Mathematics
Course Code: CSE106
Section: 2
Group: 2

Topic: CSE106 Mini-Project Report

Submitted To:
Dr. Md. Mozammel Huq Azad Khan
Professor
Department of Computer Science & Engineering
East West University

Submitted By:
Md. Miskat Hossain
ID: 2021-2-60-109
Department of Computer Science & Engineering
East West University

Nowrin Sultana
ID: 2021-3-60-027
Department of Computer Science & Engineering
East West University

Antara Sarkar Rupa
ID: 2021-3-60-056
Department of Computer Science & Engineering
East West University

Date of Submission: May 8, 2022

Information

This project is made in C language. This project is capable of generating a random undirected graph for any vertices n represented by an adjacency matrix. Also, we will be able to determine the edges and degrees of all vertices with the help of this project. This project used the handshaking theorem to determine the edges. And we will have an idea of how handshaking logic holds. We will be able to analyze the time as well.

```
Enter the number of Vertices: 5
0 0 1 0 1
0 0 1 1 0
1 1 0 0 0
0 1 0 0 1
1 0 0 1 1

0 Named Vertex has 2 Degree
1 Named Vertex has 2 Degree
2 Named Vertex has 2 Degree
3 Named Vertex has 2 Degree
4 Named Vertex has 4 Degree

Vertices: 5
Edges: 6
Total Degree: 12

If several people shake hands, the total number of hands shake must be even
For this reason the theorem is called handshaking theorem. Total Degree 12 are even.
It meant Handshaking logic holds.

Total time in Second: 4.953000s         in Millisecond: 4953.000000ms
```

Figure 01: A sample program

Analysis

We have analyzed the edge and degree for vertices 1000,2000,3000,4000 and 5000 already. Outputs are given below respectively:

```
Vertices: 1000
Edges: 250681
Total Degree: 501362

If several people shake hands, the total number of hands shake must be even
For this reason the theorem is called handshaking theorem. Total Degree 501362 are even.
It meant Handshaking logic holds.

Total time in Second: 47.907000s         in Millisecond: 47907.000000ms
```

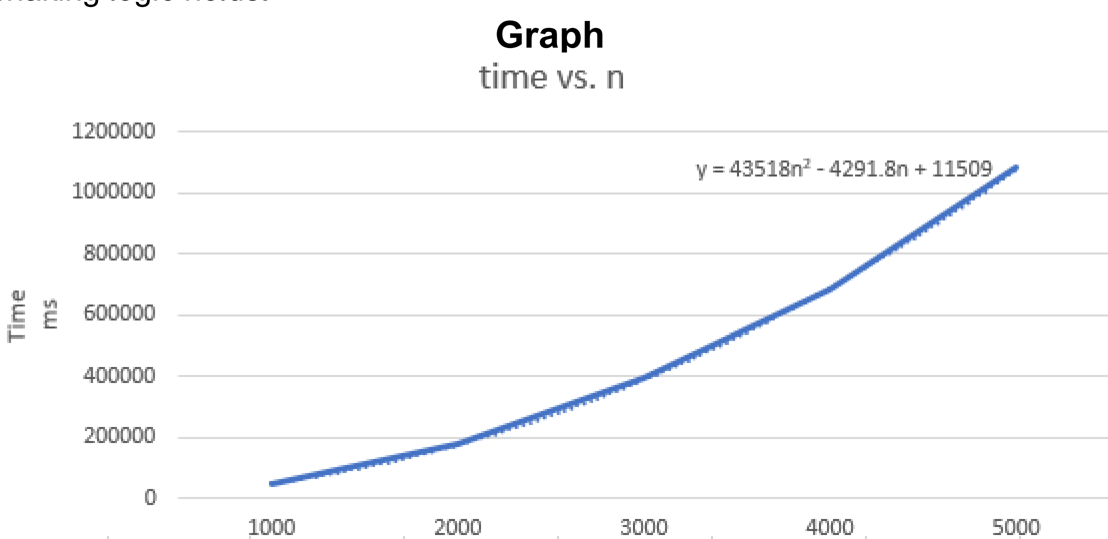
```
Vertices: 2000
Edges: 1001311
Total Degree: 2002622

If several people shake hands, the total number of hands shake must be even
For this reason the theorem is called handshaking theorem. Total Degree 2002622 are even.
It meant Handshaking logic holds.

Total time in Second: 181.742000s        in Millisecond: 181742.000000ms
```

```
Vertices: 3000
Edges: 2250589
Total Degree: 4501178

If several people shake hands, the total number of hands shake must be even
For this reason the theorem is called handshaking theorem. Total Degree 4501178 are even.
It meant Handshaking logic holds.

Total time in Second: 393.033000s        in Millisecond: 393033.000000ms
```

```
Vertices: 4000
Edges: 4000811
Total Degree: 8001622

If several people shake hands, the total number of hands shake must be even
For this reason the theorem is called handshaking theorem. Total Degree 8001622 are even.
It meant Handshaking logic holds.

Total time in Second: 682.236000s        in Millisecond: 682236.000000ms
```

```
Vertices: 5000
Edges: 6251467
Total Degree: 12502934

If several people shake hands, the total number of hands shake must be even
For this reason the theorem is called handshaking theorem. Total Degree 12502934 are even.
It meant Handshaking logic holds.

Total time in Second: 1081.741000s       in Millisecond: 1081741.000000ms
```

The theorem holds this rule that if several people shake hands, the total number of hands shaken must be even. Two degrees from two vertices connect each other like a hand. We see the sum of degrees is always even for any vertices n. If we assume degrees as hand. We can easily say handshaking logic holds.



Vertices(n)	1000	2000	3000	4000	5000
Time(t)	47907	181742	393033	682236	1081741

Hence, the time complexity of this trendline equation is in the **worst case O(n^2)**, in the **best case Ω(n^2)**, and therefore **θ(n^2) on average**.

Computational Time Complexity of the Project

First nested for loop time complexity
$n*(n+1)+1=n^2+n+1$

$n^2+n+1=O(n^2)$

```
for(a=0; a<n; a++)
{
    for(b=0; b<n; b++)
    {
        r=rand()%2;
        matrix[a][b]=r;
        matrix[b][a]=r;
    }
}
```

Second, nested for loop time complexity
$n*(n+1)+1=n^2+n+1$

$n^2+n+1=O(n^2)$

```c
for(a=0; a<n; a++ )
{
    for(b=0; b<n; b++ )
    {
        printf("%d ",matrix[a][b]);
    }
        printf("\n");
}
```

Third, nested for loop time complexity
$n*(3n+1)+1=3n^2+n+1=O(n^2)$

```c
for(a=0; a<n; a++ )
{
    ideg=0;
    for(b=0; b<n; b++ )
    {
        if(a==b && matrix[a][b]==1)
        {
            deg=matrix[a][b]+deg+1;
            ideg=matrix[a][b]+ideg+1;
        }
        else
        {
            deg=matrix[a][b]+deg;
            ideg=matrix[a][b]+ideg;
        }
    }
printf("%d Named Vertex has %d Degree\n",a,ideg);
}
```

$f(n) = n^2+n+1+n^2+n+1+3n^2+n+1$
$\quad = 5n^2+3n+3$

$f(n) = 5n^2+3n+3 <= 5n^2+3n^2+3n^2$ [$n<=n^2,1<=n^2$ for $n>=1$]
$\quad\quad =11n^2$

When $k = 1$, $C = 11$ and $g(n) = n^2$, $f(n) = 5n^2+3n+3 <= 11n^2$.
$\therefore f(n) = 5n^2+3n+3 = O(n^2)$

$f(n) = n^2+n+1+n^2+n+1+3n^2+n+1$
$\quad = 5n^2+3n+3$

$f(n) = 5n^2+3n+3 >= 5n^2$ [when $n>=0$]
$\therefore f(n) = 5n^2+3n+3 = \Omega(n^2)$ [$k=0$ and $C=5$]

Since the worst case **$O(n^2)$** and the best case **$\Omega(n^2)$** have the same time complexity, the average case is **$\theta(n^2)$**. The computational time complexity of our program and the time complexity found in **step 4** are quite the same. Both are average-case complexity and their complexity is **$\theta(n^2)$.**