CSE106: Discrete Mathematics Mini-Project

Date: May 8, 2022

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

Presented 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

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.
- This project can determine the edges and degree of all vertices.
- This project will give us a clear view of how handshaking logic holds.
- We can analysis the time as well.

How the project works

- When we enter vertices, it will generate a random undirected graph for any vertices n represented by an adjacency matrix.
- This project will determine degrees from adjacency matrix. Process given below

Sum of the raw products
Whenever diagonal element is 1, add extra 1 to it
Then sum
Sum the column of that summation

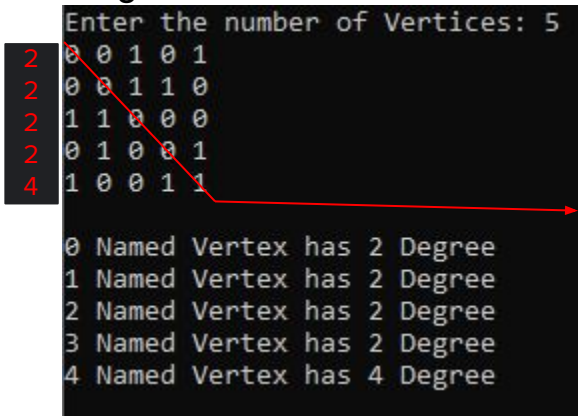


Figure 1: Project Working Method

- Once degree is determined. The edges will be determined too with the help of handshaking theorem.

Project Check

```
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 2: A sample program

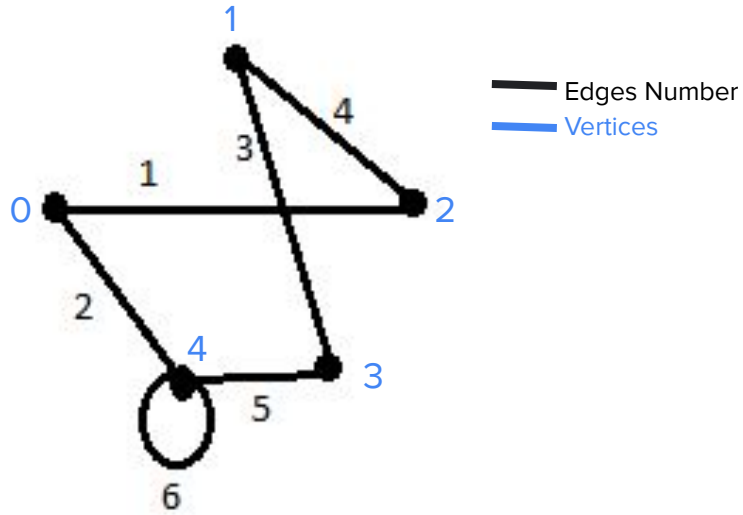


Figure 3: A undirected graph representing sample matrix

Graph showing exactly 6 edges
Program output showing 6 edges too
It meant project working perfectly

Worked with Different Vertices

When n=1000,

```
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
```

When n=2000,

```
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
```

When n=3000,

```
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.
```

When n=4000,

```
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
```

When n=5000,

```
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
```

- Generated random adjacency matrix.
- Determined the degree of adjacency matrix.
- Determined the edges with handshaking theorem.
- Handshaking theorem states that the sum of degrees of the vertices of a graph is twice the number of edges.

Handshaking Logic

Handshaking Logic: If several people shake hands, the total number of hands shaken must be even.

- From the project, we see the sum of degrees is always even for any vertices n.
- It meant every two degrees from two vertices connected each other like hand and form edges.
- Handshaking logic holds for any vertices.

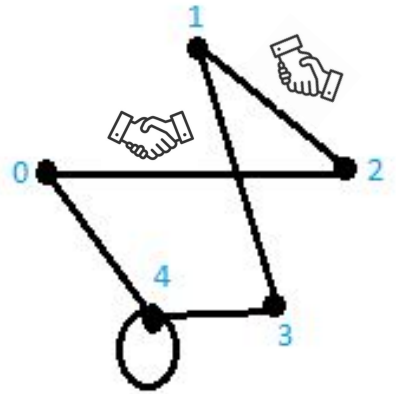
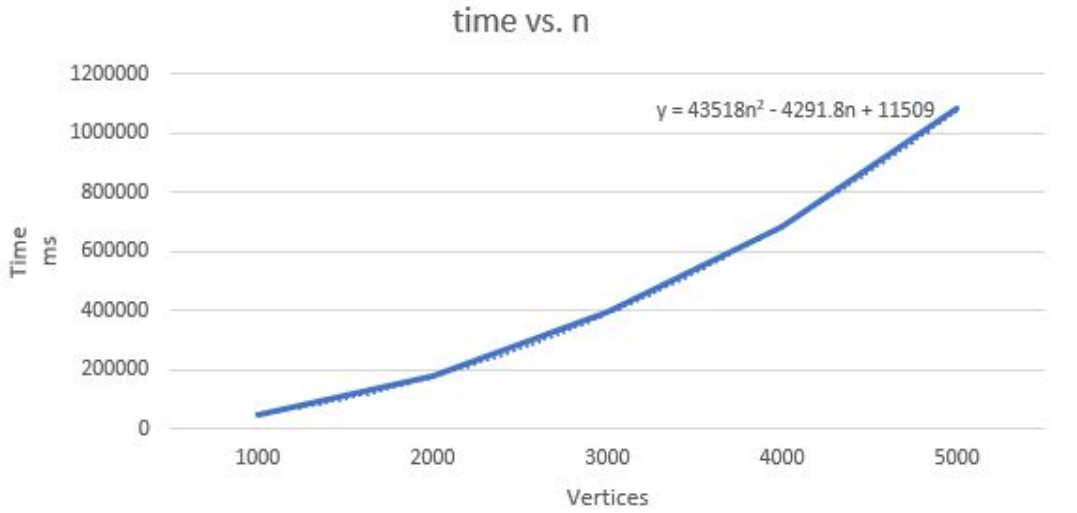


Figure 4: Handshaking Logic

Graph



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 $\Omega(n^2)$**, and therefore **$\theta(n^2)$ on average**.

Computational Time Complexity of the Project

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

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

```
15      for(a=0; a<n; a++)
16      {
17          for(b=0; b<n; b++)
18          {
19              r=rand()%2;
20              matrix[a][b]=r;
21              matrix[b][a]=r;
22          }
23      }
```

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

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

```
24      for(a=0; a<n; a++ )
25      {
26          for(b=0; b<n; b++ )
27          {
28              printf("%d ",matrix[a][b]);
29          }
30              printf("\n");
31      }
```

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

```
35      for(a=0; a<n; a++ )
36      {
37          ideg=0;
38          for(b=0; b<n; b++ )
39          {
40              if(a==b && matrix[a][b]==1)
41              {
42                  deg=matrix[a][b]+deg+1;
43                  ideg=matrix[a][b]+ideg+1;
44              }
45              else
46              {
47                  deg=matrix[a][b]+deg;
48                  ideg=matrix[a][b]+ideg;
49              }
50          }
51      printf("%d Named Vertex has %d Degree\n",a,ideg);
52      }
```

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

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

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

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

$f(n) = 5n^2+3n+3 >= 5n^2$ [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 **$\theta(n^2)$.**

Time complexity $\theta(n^2)$.

Thank You -