EAST WEST UNIVERSITY

Department of Computer Science And Engineering

Course title: CSE205 SEC-01

Report on:

Randomly generated directed graph represented by adjacency matrix.

Group members:

1. Maleha Israt Chowdhury

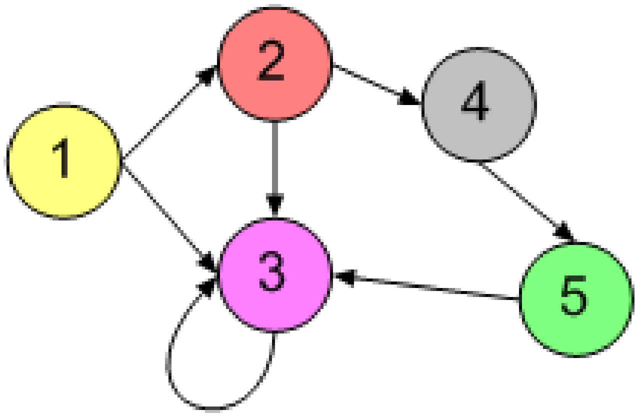
2. Shabrina Akter Shara

3. Imran Hasan

- **Problem:** Using C program randomly generate a directed graph
  represented by adjacency Matrix.

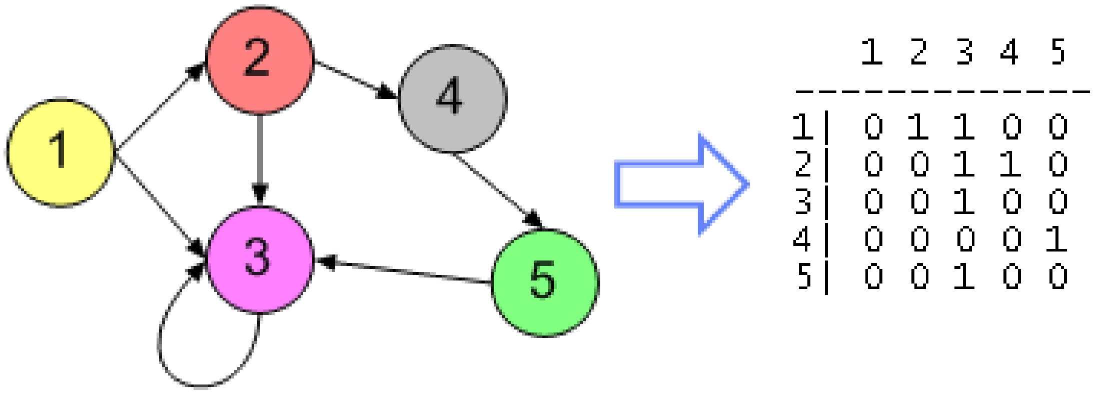
- **Directed graph:**

Directed graph consider the direction of the connection between two nodes. Below is a diagram shows a directed graph. Here node 1 and 2 connected by a directed connection and the direction is from node 1 to node 2. Node 1 can reach to node 2 but reverse is not possible.



- ## Adjacency matrix of Directed Graph:

We set element [x] [y] to 1 when there is a directed connection present between node x to y. Now this matrix may not be diagonally symmetric as element [y][x] may not be 1. Below is the output of our program.



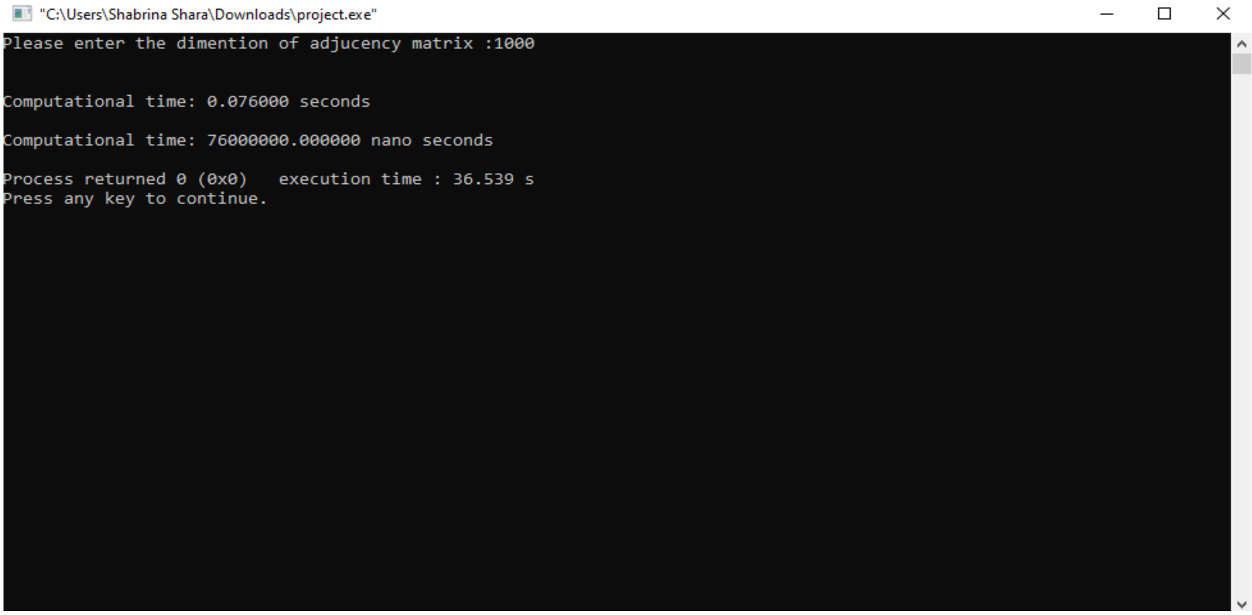
```
    1  2  3  4  5
 -------------
1|  0  1  1  0  0
2|  0  0  1  1  0
3|  0  0  1  0  0
4|  0  0  0  0  1
5|  0  0  1  0  0
```

We use this part of code for making adjacency matrix,

```
for(i=0; i<n; i++){
   for(j=0; j<n; j++)
{
      arr[i][j]=rand( )%2;
      arr[j][i]=rand( )%2;
}
```

Program outputs:

We take different input for five times and get five output with different time complexity. Outputs are:

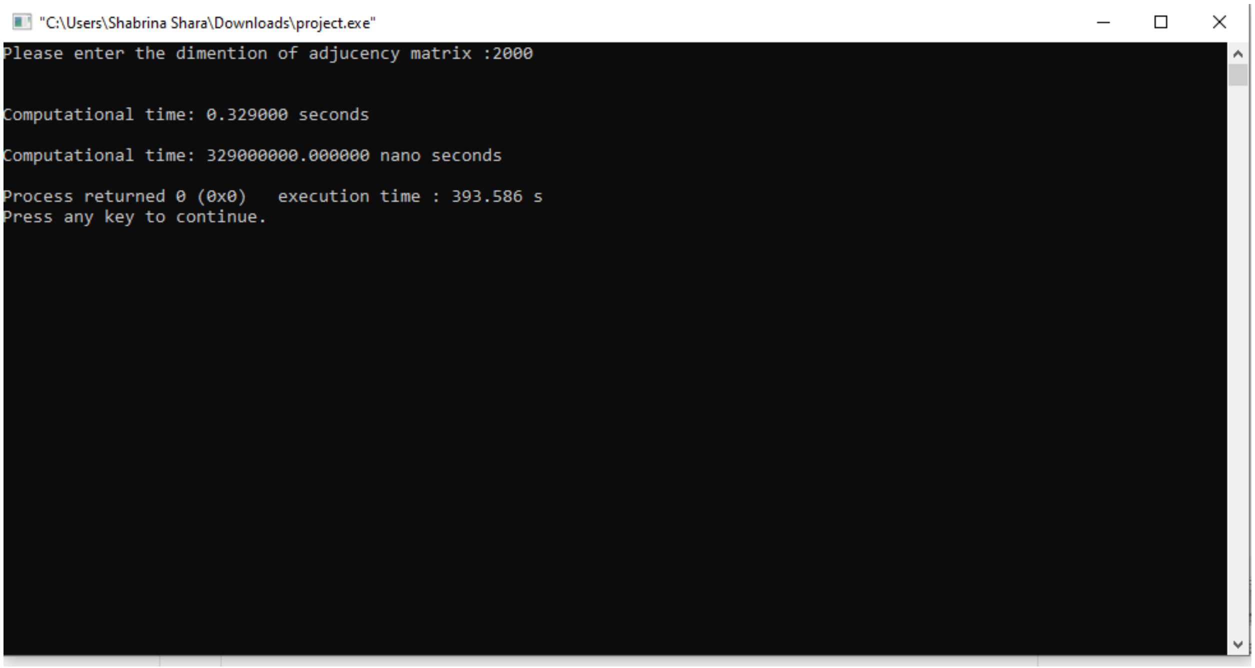


```
Please enter the dimention of adjucency matrix :1000


Computational time: 0.076000 seconds

Computational time: 76000000.000000 nano seconds

Process returned 0 (0x0)   execution time : 36.539 s
Press any key to continue.
```

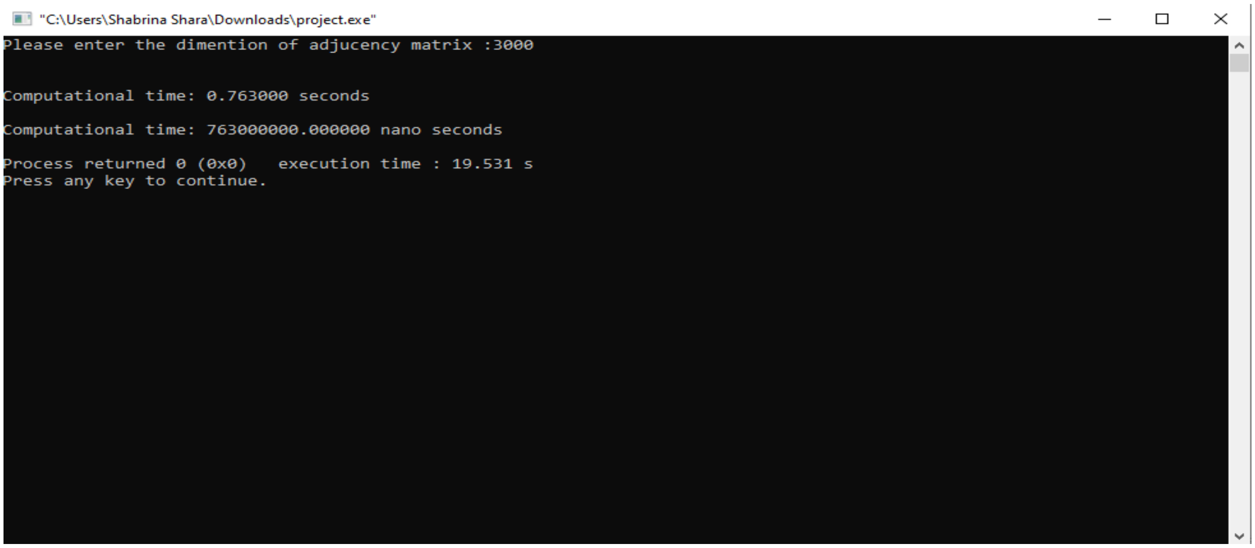


```
Please enter the dimention of adjucency matrix :2000


Computational time: 0.329000 seconds

Computational time: 329000000.000000 nano seconds

Process returned 0 (0x0)   execution time : 393.586 s
Press any key to continue.
```

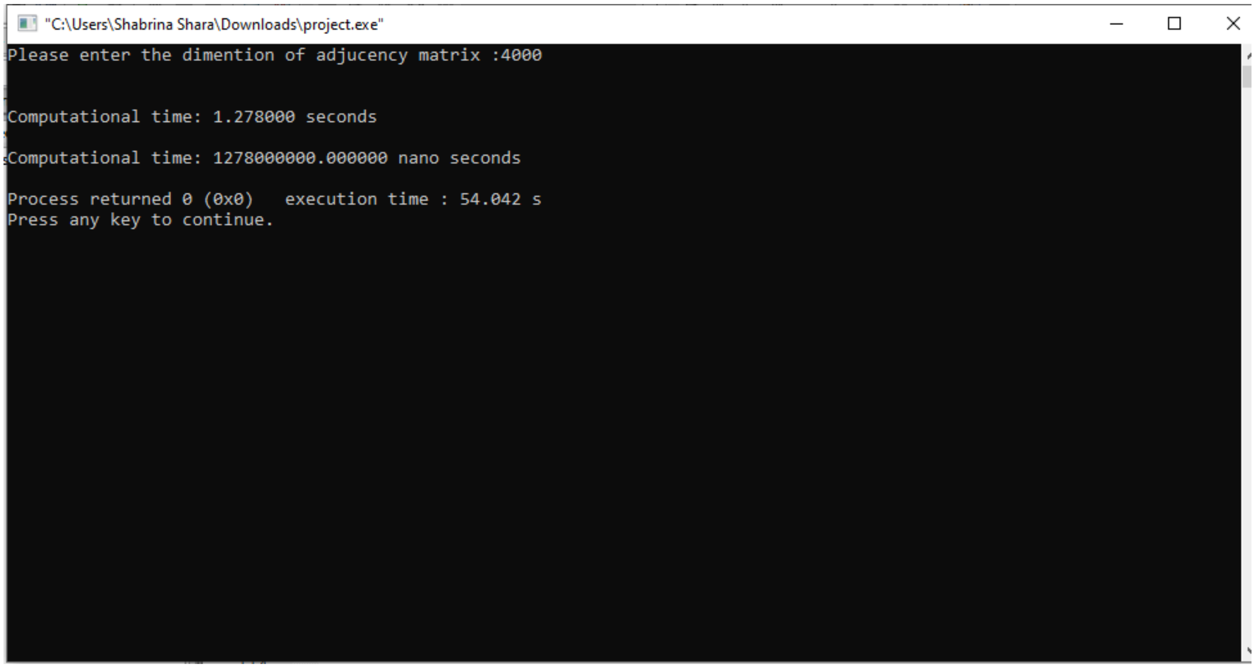


```
Please enter the dimention of adjucency matrix :3000


Computational time: 0.763000 seconds

Computational time: 763000000.000000 nano seconds

Process returned 0 (0x0)   execution time : 19.531 s
Press any key to continue.
```

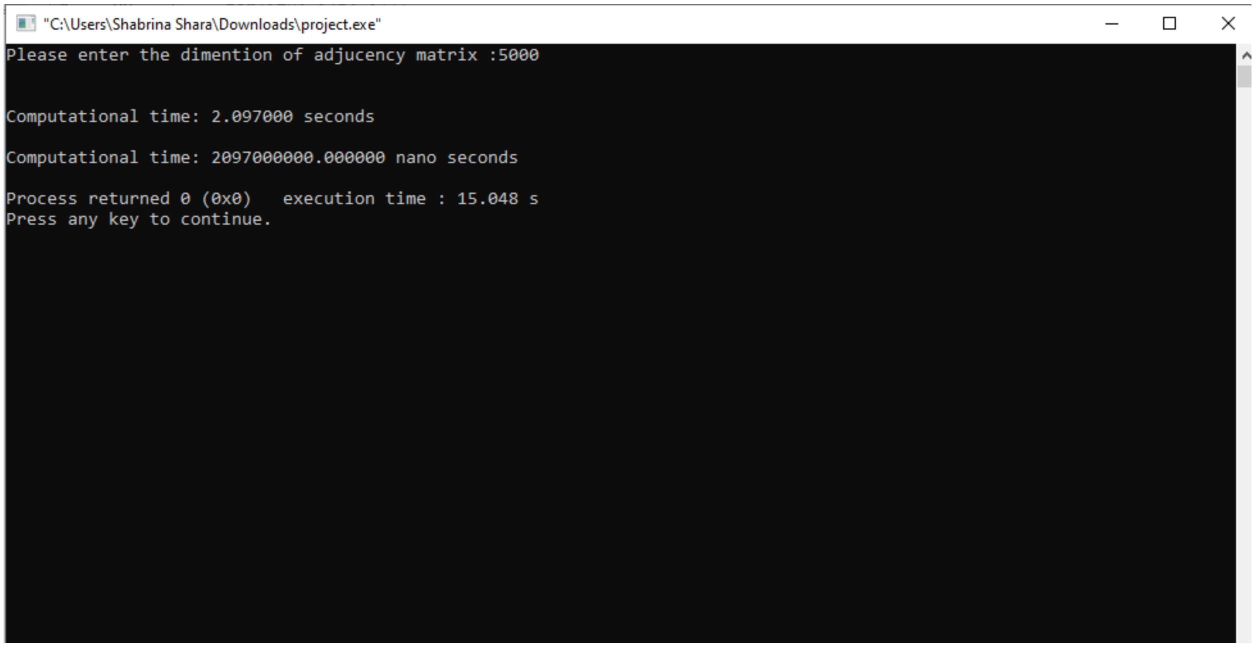


```
Please enter the dimention of adjucency matrix :4000


Computational time: 1.278000 seconds

Computational time: 1278000000.000000 nano seconds

Process returned 0 (0x0)   execution time : 54.042 s
Press any key to continue.
```



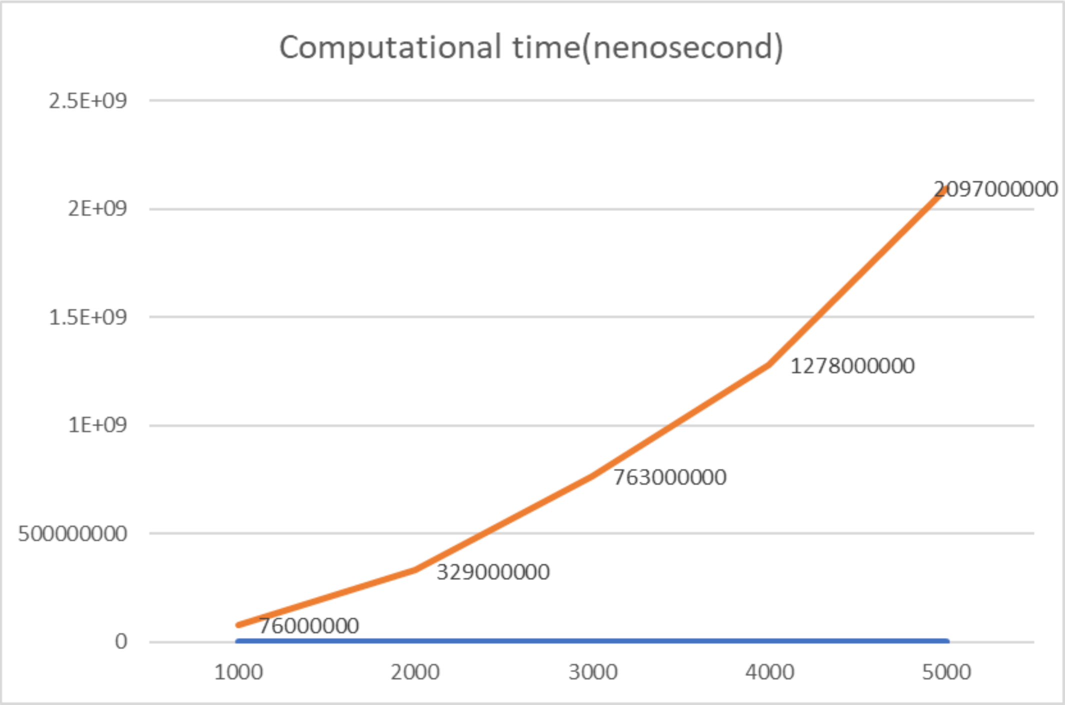
```
Please enter the dimention of adjucency matrix :5000


Computational time: 2.097000 seconds

Computational time: 2097000000.000000 nano seconds

Process returned 0 (0x0)   execution time : 15.048 s
Press any key to continue.
```

Time complexity Graph:



After pointing the output vale into excel sheet we can notice that for the output 1000 and 2000 the time complexity line is under 500000000.

When we increase the value of output in 30000 then the time complexity is under 1E+09. For the bigger output time complexity is increasing rapidly, and when the vertex number is 50000 the time complexity is upper 1.5E+09.

So we can say from graph the time complexity is near to O(n^2).

From our code, we can see that there are counting for indegree and outdegree operations. From the algorithom of our code we notice that time complexity of this code is Big O(n^2) .