**Bangladesh University of Engineering and Technology**

**Department of Computer Science and Engineering**

**CSE 208**

**Data Structures and Algorithms II Sessional**

Offline 1

Title: Running Time Comparison of Breadth-first Search (BFS) Algorithm Between Adjacency List & Adjacency Matrix Representation of Graph.

Submitted by:

Name: Tanzim Hossain Romel

Student ID: 1705069

Dept: CSE

Sec: B-1

Date of Submission: 28/02/2020

Collected data for adjacency list representation:

|  |  |  |  |
| --- | --- | --- | --- |
| V | E | Time (Adjacency List) (ns) | Time (Adjacency Matrix) (ns) |
| 1000 | 1000 | 0 | 0 |
| 1000 | 2000 | 39890 | 0 |
| 1000 | 4000 | 79860 | 3193800 |
| 1000 | 8000 | 79790 | 2692400 |
| 1000 | 16000 | 119680 | 2496500 |
| 1000 | 32000 | 179530 | 2195400 |
| 1000 | 64000 | 329130 | 2189700 |
| 2000 | 2000 | 9970 | 0 |
| 2000 | 4000 | 0 | 99700 |
| 2000 | 8000 | 99740 | 11866800 |
| 2000 | 16000 | 149600 | 11469300 |
| 2000 | 32000 | 219730 | 9770500 |
| 2000 | 64000 | 359050 | 9075700 |
| 2000 | 128000 | 648210 | 8676800 |
| 2000 | 256000 | 1246890 | 8281100 |
| 4000 | 4000 | 9980 | 99700 |
| 4000 | 8000 | 0 | 199400 |
| 4000 | 16000 | 209320 | 44381100 |
| 4000 | 32000 | 319070 | 46271700 |
| 4000 | 64000 | 508960 | 39494200 |
| 4000 | 128000 | 747990 | 38697200 |
| 4000 | 256000 | 1306510 | 34803800 |
| 4000 | 512000 | 2503460 | 33809900 |
| 4000 | 1024000 | 4897050 | 33307600 |
| 8000 | 8000 | 19940 | 0 |
| 8000 | 16000 | 10430 | 200800 |
| 8000 | 32000 | 438820 | 207245400 |
| 8000 | 64000 | 668220 | 189991800 |
| 8000 | 128000 | 966950 | 159577800 |
| 8000 | 256000 | 1556080 | 143810500 |
| 8000 | 512000 | 2712420 | 137429300 |
| 8000 | 1024000 | 5345710 | 133547600 |
| 8000 | 2048000 | 10661830 | 133339000 |
| 8000 | 4096000 | 18998720 | 131544200 |
| 16000 | 16000 | 30380 | 99600 |
| 16000 | 32000 | 39890 | 1097100 |
| 16000 | 64000 | 997320 | 832871100 |
| 16000 | 128000 | 1405910 | 753980000 |
| 16000 | 256000 | 2064470 | 637195700 |
| 16000 | 512000 | 3510920 | 572768000 |
| 16000 | 1024000 | 5983870 | 544443300 |
| 16000 | 2048000 | 11339360 | 529483700 |
| 16000 | 4096000 | 21831620 | 531275900 |
| 16000 | 8192000 | 38387480 | 520607300 |
| 16000 | 16384000 | 78769660 | 522003800 |

**Question and Answer:**

**1. What is the impact on runtime if we keep |V| unchanged and double |E| for adjacency list? Why is it so?**

*Ans:* Runtime keeps increasing gradually. From the data attached above, it can be found that runtime nearly doubles for most of the cases, which makes sense given that BFS has complexity of O(V + E).

As the number of edge increases, the number of elements that we have to check in each vertex’s adjacency list also increases, hence the increase in runtime.

**2. What is the impact on runtime if we keep |E| unchanged and double |V| for adjacency list? Why is it so?**

*Ans:* Runtime keeps increasing gradually. The reason for such increase is that the numbers of enqueue and dequeue operations will increase, which means that the loop will run for larger number of times. For larger values of V (V >> E), O(V+E) becomes almost O(V).

**3. What is the impact on runtime if we keep |V| unchanged and double |E| for adjacency matrix? Why is it so?**

*Ans:* There will be a very slight increase. Because in adjacency matrix representation, the number of iterations remains the same no matter the size of the edge. However, the number of operations inside the if-condition increases, which leads to slight increase in runtime.

**4. What is the impact on runtime if we keep |E| unchanged and double |V| for adjacency matrix? Why is it so?**

*Ans:* Runtime increases by four times. This is mainly because every time we want to find what are the edges adjacent to a given vertex 'U', we would have to traverse the whole array matrix[u], which is of course of length |V|. Hence, the total number of iterations is V^2. So, the complexity becomes O(V^2).

**5. For the same |E| and |V|, why are the runtimes for adjacency list and adjacency matrix representation different?**

**Which one is higher and why?**

*Ans:* For same E & V, adjacency matrix has higher runtime compared to adjacency list. The reason for such difference is that, regardless of whether there exists any edge between two vertices u, v, we have to iterate through each combination of u, v in adjacency matrix. Whereas, in adjacency list representation, we don’t have to iterate through combinations of vertices for which there’s no edge. Hence, the number of iterations needed for adjacency list is much smaller. The attached data also shows the same property. For that reason, BFS takes O(V+E) in adjacency list, compared to O(V^2) is adjacency matrix.