**CSCE 221-200: Honors Data Structures and Algorithms  
Assignment Cover Page  
Spring 2021**

|  |  |
| --- | --- |
| **Name:** | Priyanshu Barnwal |
| **Email:** | [priyanshu@tamu.edu](mailto:priyanshu@tamu.edu) |
| **Assignment:** | PA 5 |
| **Grade (filled in by grader):** |  |

Please list below all sources (people, books, webpages, etc) consulted regarding this assignment (use the back if necessary):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CSCE 221 Students** | **Other People** | **Printed Material** | **Web Material (give URL)** | **Other Sources** |
| 1. | 1. | 1. | 1. | 1. |
| 2. | 2. | 2. | 2. | 2. |
| 3. | 3. | 3. | 3. | 3. |
| 4. | 4. | 4. | 4. | 4. |
| 5. | 5. | 5. | 5. | 5. |

Recall that TAMU Student Rules define academic misconduct to include acquiring answers from any unauthorized source, working with another person when not specifically permitted, observing the work of other students during any exam, providing answers when not specifically authorized to do so, informing any person of the contents of an exam prior to the exam, and failing to credit sources used. *Disciplinary actions range from grade penalty to expulsion.*

"On my honor, as an Aggie, I have neither given nor received unauthorized aid on this academic work. In particular, I certify that I have listed above all the sources that I consulted regarding this assignment, and that I have not received or given any assistance that is contrary to the letter or the spirit of the collaboration guidelines for this assignment."

|  |  |
| --- | --- |
| **Signature:** | Priyanshu Barnwal |
| **Date:** | 3/18/2021 |

**Introduction:**

In this assignment, I will be creating two different implementations of graphs using adjacency matrices and adjacency lists. For each of these two graphs, I will be running two functions, known as BFS and DFS. Each of these tests will be ran four times on different types of inputs, including Cycle, Clique, 25% edge, and 75% edge.

**Theoretical Analysis:**

Theoretically, cycle and clique graphs should have a very similar runtime, and the only place the runtimes may differ is for the 25% and 75% for the list ADT because the runtime is O(V­2), and the differences in V will cause differences in the graphs as well. This means that for the adjacency list, the 25% should be a lot slower than the 75%.

**Experimental Setup:**

I am currently running this on a computer with 32GB RAM, however, Visual Studio only allocates 2 GB of processing memory for any given project. I am using Visual Studio 2019, which runs C++17 with experimental features of C++20. For my timing mechanism I used the high-resolution clock inbuilt function to accurately time how fast my program ran. I only ran each graph for one trial, but ran them 4 times by inserting cyclic, clique, 25% edge, and 75% edge values for two algorithms, BFS and DFS for three different input sizes, ranging from 100 to 10000. This makes for a total of 48 trials.

**Experimental Results:**

These graphs are all very similar, and the numbers are very close together for all four graph techniques, which is how it should be. As we can see, the 25% for each of the list arrangements end up being quite a bit higher then the 75%. Strangely, the 25% for the BFS for the list doesn’t start of higher than the 75% but ends up being much higher as the number of inputs increases. I’m not sure why that is but it could be related to the fact that we are using a linked list rather than a vector. Using a linked list causes longer insert times and longer search times than a vector which could be the main reason this is happening, but in that case it wouldn’t explain the reason that it becomes correct during the later inputs. There really isn’t anything else to say about this set of graphs, most of these are exactly as they should be. The difference is far more apparent in the DFS than the BFS for the list’s 25% which makes sense because of the way stacks work in comparision to queues. The runtime for the Stanford dataset was .42094 seconds for the list and .02279 for the matrix.