**6301.002 – Implementation of Advanced Data Structures and Algorithms**

**Long Project 0**

*Team Members:*

*Giridara Varma Elumalai*

*Srikanth Kannan*

*Praveen Erode Murugesan*

**Problem Statement:**

To find Euler’s tour in a undirected graph with time complexity O(E)

Euler’s tour: Tour that traverses all the edges of a graph exactly once

**Approach:**

Implementation of HierHolzer’s algorithm to find the Euler’s tour

**Implementation:**

Problem is solved as per the following steps:

1. Find source node of Euler’s tour.
2. Find the main tour. The edges of the main tour is maintained in the customized doubly linked list. While exploring the main tour, keep track of the vertices(incompletely processed vertices list) that has remaining edges to be processed. In addition, the index of the doubly linked list where the new tour’s edges (sub tour list) needs to be merged is also maintained.
3. Find sub tours by removing the vertex taken from the incompletely processed vertices list and having that as a source for the sub tour. Then merge the sub tour with the main tour.
4. Once all the incompletely processed vertices list becomes empty terminate the process.

**Stages of development:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Stages** | **Implementation** | **Performance on 35 MB input** | **Figured out reason** |
| 1 | Java’s linked list is used for maintaining the tours’ edges and incompletely processed vertices. | Program has not produced result. Was taking huge amount of time | Index based fetch on linked lists costed the performance |
| 2 | Java’s array list is used for maintaining the tours’ edges and incompletely processed vertices. To keep track of unprocessed edges of a vertex, a separate list is maintained as a part of each vertex | Produced the result in 50 seconds | Merging is done by traversing the edges’ list (tour list) |
| 3 | Customized doubly circular linked list to maintain tour’s edges.  Using index to keep track of position of merging of sub tour with main tour.  Using a counter to keep track of the unprocessed edges of a vertex. | Took 6-10 seconds to execute the program |  |

**Conclusion:**

The assignment helped to get deep insights about the usage of java lists and customized lists, how the execution of the program is dependent on the appropriate usage of such lists. The experiment helped to implement algorithms that will scale for large inputs without problems and taught how the linear time algorithm can make the difference in the performance when compared to the other algorithms.