**Report - Finding Euler tours**

A graph G is called Eulerian if it is connected and the degree of every vertex is an even number. It is known that such graphs always have a tour (a cycle that may not be simple) that goes through every

edge of the graph exactly once. Such a tour is called an Euler tour.

This project finds an Euler tour in a given graph.

Initial code base: [Graph classes](http://www.utdallas.edu/~rbk/teach/2016f/java/graph.zip) (Provided by Professor)

**Files used:**

Driver.java - file to read the input graph, call method to find a tour in it, verify for correctness and print output

Vertex.java - file to define a vertex and methods for it

Edge.java - file to define an adge and methods for it

Graph.java - file to break a given graph, merge the subtours and verify it

circularList.java - file to implement a circular list and merge the subtours

**Input format**

Through command line args[0] takes the name of a file from which input is read.

Input can also be read through standard input (console).

The graph is assumed to be simple.

**Output format**

The output is printed to standard output (console).

If the graph has no Euler tour, it prints :"Graph is not Eulerian" otherwise prints the nodes of an Euler tour, starting at node 1.

|  |  |
| --- | --- |
| **Sample input**  6 10  1 2 1  1 3 1  1 4 1  1 6 1  2 3 1  3 6 1  3 4 1  4 5 1  4 6 1  5 6 1 | **Sample output**  1  2  3  6  4  5  6  1  3  4 |

**Results:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Files | Time Taken for each method | | | |
| Reading | Breaking graph into tours | Stitching the sub tours | Verifying the tour |
| mp1\_input1 | 91 msec | 19 msec | 0 msec | 1 msec |
| mp1\_input2 | 364 msec | 114 msec | 1 msec | 4 msec |
| mp1-big | 51694 msec | 2693.486 sec | 110 msec | 468 mse |
| Memory consumption | | | | |
| mp1\_input1 | 4mb/123mb | 4mb/123mb | 4mb/123mb | 4mb/123mb |
| mp1\_input2 | 14mb/123mb | 14mb/123mb | 14mb/123mb | 14mb/123mb |
| mp1\_big | 479mb /923mb | 615mb/ 923mb | 615mb/ 923mb | 624mb / 923mb |