**Group Details: G24**

**\* Nikhil Sathe, Ankush Raizada, Kunal Krishna, Yogesh Darji**

Input: Graph G (V,E) – Complexity O(|E|)

Output: Euler tour, one of the possible tours

Algorithm:

1. Calculate in-degree of all nodes

2. If all in-degrees are even, search for Euler tour by following below steps:

1. Mark any node as source and add its first edge connecting to its first neighbor

2. From that neighbor, go on traversing the graph

a. Check if unvisited neighbors are more than one don't go back to source

b. If unvisited neighbor is only source then go to source and now mark source as unvisited neighbor of previous source and repeat the process

else

Given graph is does not have an Euler tour.

Observations:

|  |  |  |
| --- | --- | --- |
| #Edges | Time with printing tour(in ms) | Time without printing tour(in ms) |
| 10 | 2 | 0 |
| 1050 | 149 | 4 |
| 5506 | 430 | 11 |
| 5249924 | 458884 | 45714 |