CS 6301.002

Implementation of advanced data structures and algorithms

Long Project 0: Finding Euler tours

G-07: Salil Kansal, Twinkle Sharma, Sujit Sajja

Report

Methods used:

- printEuler(Graph):void:
  The main method which checks the graph for connectivity and also calculates the no of odd degree vertices. It also prints the euler tour and euler path after running the algorithm.
- findEuler(Vertex):List<Edge>
  It finds a euler tour/ path starting from the src vertex.
- isConnected(Graph):boolean
  Return true if the graph is connected
- getOddDegreeVertex(Graph):List<Vertex>
  Returns the list of vertices with odd degrees
- BFS(Vertex):void:
  Runs BFS on the graph from the Vertex src.
- verifyTour(Graph,List<Edge>,Vertex):Boolean
  It verifies if the tour is correct by traversing it.

Problems faced:

- Getting the next unvisited edge in constant time
- Stitching the subpath into the main path
- All the vertices were not being traversed in the original implementation

Design Descisions:

- For next unvisited edge, earlier we were traversing the adj list from starting and returning the first unvisited edge. Later on we added an index currentUnvisitedEdge below which all edges are visited. This improved the running time drastically.

Test results:

- All input files provided are giving the required output.
- For the big data set it takes around 2 seconds.
- References:

StackOverflow

Wikipedia