Mandatory Project-3

Manoj Kumar Ellanti

netId:mxe150630

* BinaryHeap.java is used to implement the indexed part of the Priority Queue. sizeFilled is used in the BinaryHeap to keep note of the elements filled in the heap.
* sizeFilled is update whenever an element is added or removed from the heap.
* Indexes are updated for all the vertices according to their positions in the input array.
* Indexes are updated in the percolateup(), and percolateDown() method to all the affected vertices.
* buildHeap() is called in the constructor , so as to construct the heap at the beginning itself, code for the index initialization is written in the buildHeap().
* prim1(), and PrimMst() in the MST.java is used to write the prim's Minimum spanning tree calculation using Priority Queue of Edges, and Priorityqueue of Vertices respectively.
* class EdgeComparator, class Comp is coded for the priority queue of edges and vertices respectively.
* DijkstrasShortestPaths(), is used to code for the Shotestpath calculation using Dijkstras algorithm from a given source.
* travellingSalesMan(Graph g,Source s) is coded in ShortestPath.java.
* first part of this method contains the code for the construction of MST using rim2 algorithm. This method calls modifiedDfs() to calculate the distance. Also Children arrayList is added to vertex to keep track of all the children of a given node in the MST. So that they are used while calculating the path. The children arrayList is updated in the travellingSalesMan() with minor changes.
* modifiedDfs(Vertex s, Vertex PossibleCousin, Vertex mainSource) is a modification of DFS, where distances are added(s.d which is calculated in MST updation, ) among vertexes of s using DFS till leaf nodes are found. Once Leaf node is found, The distance is now incremented with the edge weight between s and it's possibleCousin.
* Once leaf nodes are found the same method is called with s=PossibleCousin. PossibleCousin at every level is It's Sibiling or, it's Parent's sibling or it's ancestor's sibling going on hierarchically, going upwards. It is updated using it's parent's or ancestor's Children arrayList.
* This method is repeated till, s and PossibleCousin both become's null. Then an edge is verified between the leaf that is iterated at the end and the mainSource vertex, and it's weight is added to the distance to complete the cycle.