CSC700-34: Project 7B Prim’s MST (JAVA)

Shawn Yang

Due Date: Nov.30, 2017

Program implemented as the algorithm steps below (provided by Prof. Phillips)

Algorithm Steps:

0. numNodes 🡨 get from input file

Allocate and initialize all the members in PrimMST class accordingly

Step1 to Step 3 is to make a list of input edges in ascending order so that for the subsequence of

finding a min cost edge would be easy.

1. <Ni, Nj, edgeCost> 🡨 read from input file

newEdge 🡨 create an undirectedEdge and fill with <Ni, Nj, cost>

insertEdge ( newEdge, edgeListHead) // Using insertion sort to insert newEdge into the linked list of

edgeListHead (in ascending order by the cost)

graphNodeArray[Ni]++

graphNodeArray[Nj]++

2. printList (edgeListHead)

3. repeat step 1 to step 2 until the input file is empty

4. nextEdge 🡨 findAndRemoveNextEdge() Check to see if ni and nj are in same set, if yes, move on to next in

the list without removing it. If not in same set, return the edge and delete it from edgelisthead

5. repeat step 4 until Ni and Nj are in different sets

6. pushEdge(nextEdge, MSTofG) // push nextEdge in the front of MSTofG

totalMSTCost += the cost of nextEdge

if Ni is in setA,

move2SetA(Nj, setA) // move Nj from setB to setA

else

move2SetA(Ni, setA) // move Ni from setB to setA

// now, Ni, Nj should be in setA

Note: while changing set, make sure to change all the nodes in the same set to new set as well.

printSet(inWhichSet) // print the inWhichSet array to argv[3] (with proper heading)

7. printList(MSTofG) // print up to the first 10 edges of MSTofG) to argv[2] file (with proper heading)

8. repeat step 4 – step 7 until setB is empty. // inWhichSet are all 1

9. output the entire MSTofG and the totalCost to outfile1, argv[1]

10. close all files.