CSC 323-32: Project 5 <Prim’s MST > (C++)

Preston Peck

Due date: Nov. 10, 2016

\*\*\*\* Algorithm steps for Prim’s Algorithm:

Step 0: N 🡨 get from input file

0.1: initialize all objects in PrimMST class accordingly

Step 1: newEdgeNode 🡨 read each <Ni, Nj, cost> from input file to create an edgeNode

1.1: insert newEdgeNode into edgeList (in ascending order by the cost)

1.2: graphNodeIDarray[Ni]++

graphNodeIDarray[Nj]++

Step 2: repeat step 1 until the input file is empty

Step 3: print graphNodeIDarry to output-2 file

3.1: print edgeList (only the first 10 edges) to output-2 file

Step 4: k 🡨 find the first none zero graphNodeIDarray[k],

4.1: newGraphNode 🡨 create a new graphNode for k

4.2: insert newGraphNode into SetA

Step 5: k 🡨 get the next none zero graphNodeIDarray[k]

5.1: newGraphNode 🡨 create a new graphNode for k

5.2: insert newGraphNode in the front of setB

Step 6: repeat step 5 until reach the end of graphNodeIDarray

Step 7: newEdge 🡨 remove the front edge, say <Ni, Nj,cost>, from the edgeList

if Ni and Nj are in the same set, discards newEdge

7.1: repeat step 7 until Ni and Nj are one in setA and the other in setB

Step 8: insert newEdge in the front of MSTofG

8.1: add the cost in the newEdge to totalCost

8.2: move the graphNode (Ni or Nj whoever in setB) from setB to setA

Step 9: print setA to output-2 file

9.1: print setB to output-2 file

9.2: print MSTofG to output-2 file

Step 10: repeat step 7 – step 9 until setB is empty

Step 11: output MSTofG and the totalCost to output-1 file with proper heading, one edge with

cost per text line.

Input

12

6 4 13

12 7 13

6 12 17

10 12 27

9 10 24

2 4 18

9 11 15

3 2 11

5 7 25

1 6 13

8 6 22

9 8 32

8 10 31

5 4 26

4 3 13

1 2 16

1 11 26

3 5 14

6 7 12

Output

Minimum Spanning Tree

12

3 2 11

6 7 12

4 3 13

6 4 13

1 6 13

3 5 14

9 11 15

6 12 17

8 6 22

9 10 24

1 11 26

Total Cost is: 180

Graph Node 1D Array:

3 3 3 4 3 5 3 3 3 3 2 3

Edge List:

6 7 12

4 3 13

1 6 13

6 4 13

3 5 14

9 11 15

1 2 16

6 12 17

2 4 18

8 6 22

Set A: 6 1

Set B: 12 11 10 9 8 7 5 4 3 2

Minimum Spanning Tree:

1 6 13

Set A: 7 6 1

Set B: 12 11 10 9 8 5 4 3 2

Minimum Spanning Tree:

6 7 12

1 6 13

Set A: 4 7 6 1

Set B: 12 11 10 9 8 5 3 2

Minimum Spanning Tree:

6 7 12

6 4 13

1 6 13

Set A: 3 4 7 6 1

Set B: 12 11 10 9 8 5 2

Minimum Spanning Tree:

6 7 12

4 3 13

6 4 13

1 6 13

Set A: 2 3 4 7 6 1

Set B: 12 11 10 9 8 5

Minimum Spanning Tree:

3 2 11

6 7 12

4 3 13

6 4 13

1 6 13

Set A: 5 2 3 4 7 6 1

Set B: 12 11 10 9 8

Minimum Spanning Tree:

3 2 11

6 7 12

4 3 13

6 4 13

1 6 13

3 5 14

Set A: 12 5 2 3 4 7 6 1

Set B: 11 10 9 8

Minimum Spanning Tree:

3 2 11

6 7 12

4 3 13

6 4 13

1 6 13

3 5 14

6 12 17

Set A: 8 12 5 2 3 4 7 6 1

Set B: 11 10 9

Minimum Spanning Tree:

3 2 11

6 7 12

4 3 13

6 4 13

1 6 13

3 5 14

6 12 17

8 6 22

Set A: 11 8 12 5 2 3 4 7 6 1

Set B: 10 9

Minimum Spanning Tree:

3 2 11

6 7 12

4 3 13

6 4 13

1 6 13

3 5 14

6 12 17

8 6 22

1 11 26

Set A: 9 11 8 12 5 2 3 4 7 6 1

Set B: 10

Minimum Spanning Tree:

3 2 11

6 7 12

4 3 13

6 4 13

1 6 13

3 5 14

9 11 15

6 12 17

8 6 22

1 11 26

Set A: 10 9 11 8 12 5 2 3 4 7 6 1

Set B:

Minimum Spanning Tree:

3 2 11

6 7 12

4 3 13

6 4 13

1 6 13

3 5 14

9 11 15

6 12 17

8 6 22

9 10 24

1 11 26