Lab Problem 1

COP3503 assigned Mar 17, 2021
Michael McAlpin, Instructor due April 8, 2021

1 Goal

Find the **Minimum Spanning Tree** *(MST)* in a graph using **Prim's** algorithm.

2 Problem

1. Input data will be in a text file that contains the graph as an *Adjacency Matrix*. This input file's name will be specified in the command line as the first parameter. The file format is specified as follows:
   (a) The first record is a single integer with the maximum number of vertices. (*This is a one-relative number as the vertices are numbered from 0 to Maximum Number Of Vertices - 1.*)
   (b) The second record is a single integer with the number of edges in the input file.
   (c) Subsequent records contain the edge's specfic data, as follows:
      i. The first element is an integer representing the one vertex of the edge.
      ii. The second element is an integer representing the other vertex of the edge.
      iii. The last element is an floating point number representing the weight of the edge.
2. The expected output will be the *MST* for the given graph to include the *total weight of the MST*.

2.1 Design Approach

The design of the ***Prim MST*** should be based on the *psuedocode* shown in Lecture 13, and also shown below.

```
MST-PRIM(G,w,r)
1.  for each u ∈ G.V
2.     u.key = ∞
3.     u.pi = NIL
4.  r.key = 0
5.  Q = G.V
6.  while Q ≠ ∅
7.     u = EXTRACT-MIN(Q)
8.     for each v ∈ G.Adj[u]
9.        if v ∈ Q and w(u,v) < v.key
10.          v.pi = u
11.          v.key = w(u,v)
```

Note that there is also a test input file named, `lec13Prim.txt` provided in the test data.

3 Submission

via WebCourses

1. The single Java source file, named `Lab01.java`.
2. IMPORTANT Make sure that your submission has your name in a comment block at the very front of the file. **Make sure that your team mate's name is also in that comment block.** The two team members **must** submit identical java files in order to receive full credit.
   - Use the comment block shown below as your template.

```
/* COP3503 - CS II
 * Lab 01 - Prim's algorithm
 * Submitted by:
 *     Fred Flintstone
 *     Barney Rubble
 */
```

4 Testing

There are four input files supplied with the assignment:

1. **lec13Prim.txt** which contains 5 vertices and 6 edges.
2. **in8v16e.txt** which contains 8 vertices with 16 edges.
3. **in250v1273e.txt** which contains 250 vertices with 1273 edges.
4. **in1Mv758Ke.txt** which contains $1,000,000$ vertices with $\sim 758,000$ edges.

The test script, `lab1Test.sh` is included in the ZIP file. It compares student outputs to the expected outputs in a correspondingly named `Base` file. *NB: Use this as guidance for the total weight of the MST. Some variations are allowed, as there may be subtleties in the* ***MinQ*** *functions which might change output order.*

5 Sample output

Sample outputs are included in the assignment ZIP file[1]. They are named to correspond to the input file's name. The sample below is derived from Lecture 13's Prim Example problem.

```
~/labs/L1/code/tst $ java Lab01 lec13Prim.txt
0-2 0.20000
0-1 0.30000
1-4 0.20000
2-3 0.40000
1.10000
~/labs/L1/code/tst $
```

6 Grading

Grading will be based on the following rubric:

Table 1: Grading Rubric

Percentage	Description
-100	Cannot compile on *Eustis*
-100	Cannot read input files as specified in the command line.
-100	Does NOT specify team member name or *no explicit statement that this is a solo submission.*
- 50	Does not output the vertices and edges for the *MST*.
- 50	Cannot calculate **MST** for given input file.

[1]The output for the input file named **in1Mv758Ke.txt** is not included