ECE 250 - Project 4
Minimum Spanning Tree using Kruskal's Algorithm
Design Document

Ayushi Tiwari, UW UserID: attiwari

Apr 3rd, 2020

1. Overview

Class 1: Node

Description: Node class has information of the nodes of the linked list.

Member Variables:
1. Integer data;
2. Node type pointer next (points to the next node);
3. linkedList type pointer info;

Class 2: linkedList

Description: This class helps to build sets through linked list and store information.

Member Variables:
1. Node type head
2. Node type tail
3. Integer size

Class 3: disjointSets

Description: This class helps in making sets or finding sets using liked lists and used in the kruskal's algorithm.

Member Variables:
1. Node type newNode
2. linkedList type Set
3. integer counter
4. integer numberOfSets

Member Functions:
1. void makeSet(int i);
2. int findSet (int n);
3. void unionSet (int u, int v);

Class 4: EdgeHelp

Description:
It is a class used for checking if the two graphs are connected or not and also for saving the values of vector.

Member Variables:
1. Integer u;
2. Integer v;
3. Double edgeweight

Member Functions (operations):
1. EdgeHelp(int v1, int v2, double w)

2. Constructors/Destructors

Class linkedList (Constructor and Destructor):
They are used for making new arrays of node types to store the information and the destructor is used to deallocate the memory of all the pointers.

Class Graph (Constructor and Destructor):

It is used for dreating the graph by using loops and setting the values to null in the beginning.

3. Test Cases

Test 1: Create a graph, add some information delete it and then return the edge count and clear it

Test 2: Create graph, add nodes clear it, add nodes again and then create an MST.

Test File Example 1:

```
n 4
i 0;1;10
i 0;2;6
i 0;3;5
i 2;3;4
i 1;3;15
degree 0
edge_count
mst
```

Test File Example 2:

```
n 10
i 0;1;0.3
i 2;4;1.2
i 3;2;2
i 2;0;-0.7
d 2;3
d 1;2
d 5;0
mst
```

4. Performance

The create graph function resizes the matrix, so its time complexity is O(V^2) where V is the number of vertices because it's a VxV matrix.
The insert and delete function run in linear time as are setting the value of one element in the matrix either to a number or 0.
The degree function consists of a for loop which runs through the row of that vertex in the matrix to count the edges, so it has a time complexity of O(V).
The edge count returns a variable in linear time.
Clear function runs 2 loops in O(V^2) time as it has to delete the whole matrix.

5. References

https://www.tutorialspoint.com/prim-s-algorithm-simple-implementation-for-adjacency-matrix-representation-in-cplusplus
http://www.stargroup.uwaterloo.ca/~ece250/materials/notes/Lecture27-MST-PartB.pdf
https://github.com/samiurrahman98/ECE250/tree/master/Lab4
https://www.geeksforgeeks.org/prims-algorithm-simple-implementation-for-adjacency-matrix-representation/