Minimum Spanning Tree Reflection

Charles Bailey

04/25/2015

CMPT 390

The following program is an implementation of Prims and Kruskals algorithms for Minimum Spanning Trees on Undirected Graphs. This was the only assignment from CMPT 306 that involved programming, as other programming assignments in the class were labs or included with the Midterm and Final tests. Writing this program involved the Computer Science learning goal of developing effective problem solving skills. The disjoint Set program was already written for us and the students were responsible for implementing the functions that implemented Prims and Kruskals.

This program involved having to pick an appropriate data structure to represent an adjacency List, the visited edges, the minimum spanning tree, and the priority queue. The adjacency list is a dictionary data structure that held the adjacent edges for the current edge. The purpose of having the adjacent edges was to add the adjacent edges to the priority queue. The visited edges is a list that keeps track of which edges have already been checked by the algorithm. The minimum spanning tree was a list that contained the edges that are included with the minimum spanning tree of the undirected graph. The priority queue was a list that can be used with the heapq functionality of python and can give a priority to items based on numerical value. One of the many mistakes I made when first using the priority queue was not having the right value listed first on the edge. I tried having a node as the first object in a tuple, but when I switched it to the cost of the edge, the priority queue started implementing the algorithm correctly. This program also fits the problem solving learning goal because I often had to look up information on how the algorithms work, and then write the code to implement the various steps of the program. While implementing Prims was easier as much of the code was written for the function already, Kruskals was harder to implement. I did not need to use much of the code that was already written, but I had to find out how Kruskals worked, and then think of a pseudo code approach to how to solve it. I also had to implement two functions from the DisjointSet program and figure out where they would be used. I often had some compilation issues as the find function needed the root of the edge, but the current function had the roots as strings. The find function would only accept integers, and therefore the root needed to be converted into an integer value. I learned from looking online that python had an ord function that could convert a string value into a Unicode value. Although the ord function worked for the roots, the priority queue still was not returning the values in the minimum spanning tree. I had to subtract 96 from the Unicode values while implementing the find function, and then use the union function to add the edge to the minimum spanning tree. My final issue was indentation with python and where to put certain code. Once the code was put in the right place, both kruskals and prims algorithms were working. The assignment taught me about what data structures are appropriate to use in python, how to implement an algorithm, as well as looking to outside sources for how to perform certain actions in python such as converting a string character to an integer.