CSCI203

assignment 3

Student name: Weicheng Yin

Student number: 5009819 | Email ID: wy157

Introduction:

Minimum spanning tree algorithm is used in many application scenarios. Because according to this algorithm, it will always generate the least cost. The scenarios such like network design make the use of this algorithm. Suppose one company has several offices in different locations. The company want to make the cost least to connect them using phone line. In this scenarios, we could think offices as nodes, phone line as edges. The use the MST algorithm to implement it. In this assignment, our background is to use two different algorithms to find the minimum spanning tree. The purpose is to analysis which algorithm is more efficient.

Methods:

In this assignment, I used two different algorithms which are brute force and prim algorithm. And I used a 2-D array as the data structure to represent the weight matrix.

-Brute force:

For the brute force, the general idea for this algorithm is to find all the possible minimum spanning trees and then find the minimum cost of all the possible spanning trees. For example, if the node number is 4, it could be easily calculated that the total edges number for a fully connected graph is 4*(4-1)/2 which is 6. And if the node number is 4, the total edges for its' spanning tree is 3. So I will use a 6-digit sequence to represent whether the edge is selected or not. For example, if the bit sequence is like 001011, it represents a spanning tree. And the edges contained in this tree is edge (1,4), (2,4), (3,4). I will represent the edge in this order: (1,2), (1,3), (1,4), (2,3), (2,4), (3,4). After I found all the possible combinations which contains three 1s in this situation, the next step is check the connectivity for the graph. Because maybe it will contain some loops, and that's not a tree structure. After I found all the possible spanning tree, I will find the tree contains the least cost, that's the minimum spanning tree.

-Prim:

For the prim algorithm, it will start from node 1, then consider all the nodes connected with node 1, find the least cost, put the node with the edges with node 1 in to the connected node set. Then consider all the nodes connected with the connected node set, find the least cost and add the new node into the connected node set. After node number – 1 steps, I could find the minimum spanning tree.

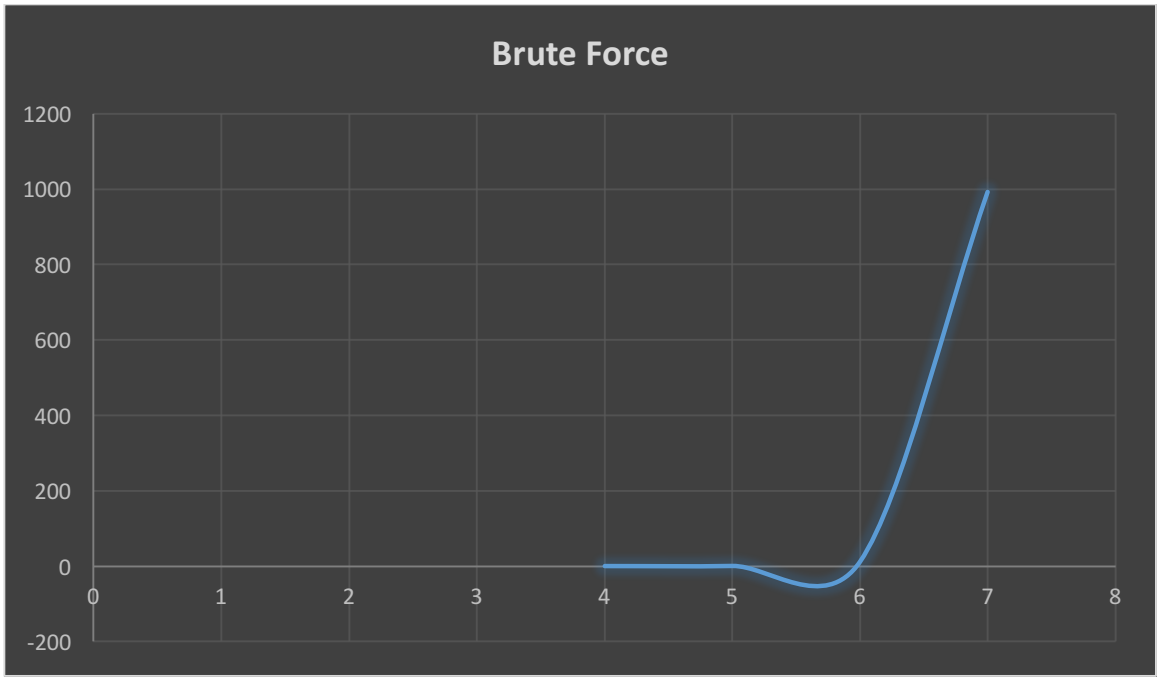
Result analysis:

I ran four data sets to test my program, each of them contains 4, 5, 6, 7 nodes.

-brute force

The following plot is for the brute force algorithm. The x coordinate is the node number for the test, the y coordinate is the time for the node number. As you can see from the plot, while the node number increased, the time with that number nodes will increase significantly. But there's a negative area in the plot. It is because I used Microsoft the draw this plot, and choose the round plot. To make it connected, the plot will go down and produce the negative part. Here's table contains the data set.

Node number	Time(milliseconds)
4	0.049
5	0.351
6	11.534
7	992.985

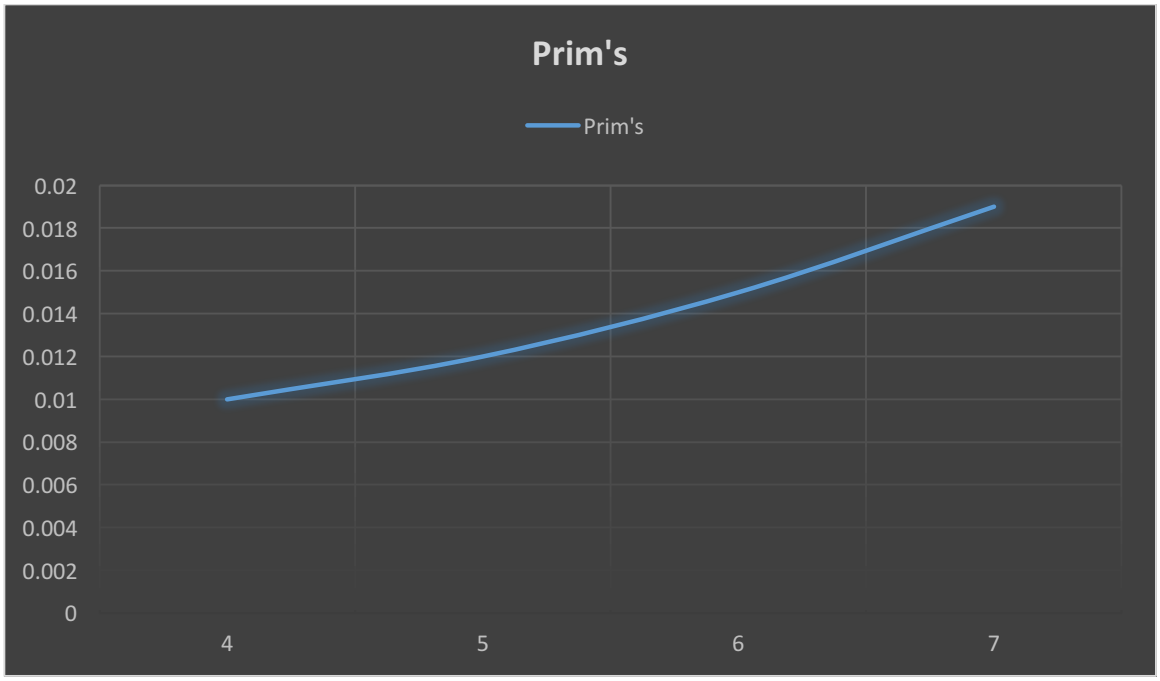


the time complexity for the brute force algorithm could be calculate like this: if the node number is N, first we need to find all the possible combination of the spanning tree, which could cost n!/(c!*(n-c)!) time. Then we need to loop through all the spanning tree to find the least cost of the spanning tree. Which will cost n steps. So the time complexity in big O notation is: O(n!).

-Prim's:

The following plot is for the Prim's algorithm. For this algorithm, I also ran 4 data sets test. Here's the table for the time and the node number:

Node number	Time(milliseconds)
4	0.01
5	0.012
6	0.015
7	0.019



Compare to the plot we got for Brute Force algorithm, the line grows less not too fast while the node number increased. And the time complexity for the Prim's algorithm by using adjacency matrix could be represent like this: O(v^2).

Conclusion:

After I finish this assignment, I have a deep understanding to the MST algorithm. The brute force algorithm will cost more time and space to find the minimum spanning tree. And the prim's algorithm is efficient compare to the brute force algorithm. In the future, if I'm looking some solution like the network design, I will use prim's algorithm to implement it.

Reference:

Combinatorial Optimization and Applications: 4th International Conference, COCOA 2010, Kailua-Kona, HI, USA, December 18-20, 2010, Proceedings, Part I (Lecture Notes in Computer Science). 2010 Edition. Springer.

IEEE Xplore Document - Shortest connection networks and some generalizations. 2016. *IEEE Xplore Document - Shortest connection networks and some generalizations*. [ONLINE] Available at:http://ieeexplore.ieee.org/document/6773228/?arnumber=6773228. [Accessed 29 September 2016].

http://www.nbs.sk/. 2010. *Minimum Spanning Trees⁎*. [ONLINE] Available at:http://www.nbs.sk/_img/Documents/_PUBLIK_NBS_FSR/Biatec/Rok2013/07-2013/05_biatec13-7_resovsky_EN.pdf. [Accessed 30 September 2016].

http://www.nbs.sk/. 2011. *Minimum Spanning Tree Application in the Currency Market*. [ONLINE] Available at: http://www.nbs.sk/_img/Documents/_PUBLIK_NBS_FSR/Biatec/Rok2013/07-2013/05_biatec13-7_resovsky_EN.pdf. [Accessed 30 September 2016].