Project 2

Ang Li (G40625167) | ang\_li@gwu.edu

# Problem Statement

Find a minimum spanning tree of the given graph by implementing the Kruskal’s Algorithm.

# Implementation Characteristics

First, sort edges by weight. Next, initialize counter i=1. Then, initialize an empty tree T. While the number of edges in tree T is less than n-1, and if the adding edge does not create a cycle, add the edge into tree T.

Sorting the edges costs O(m log m) time, and inside the while loop, we use 2 Find Operations to check if the adding edge will create a cycle. This step can be executed once per edge, so it requires O(m log n). When adding an edge, we use 1 Union Operation and there are n-1 edges. It requires O(n log n). And total is O(m long m).

# Experimental Analysis

## Programe Listing

while (i != edge) {

if (union\_tree(edge\_tag[i], parent, child)) {

cout << ("v" + std::to\_string(edge\_tag[i].start))

<< "-----"

<< ("v" + std::to\_string(edge\_tag[i].end))

<<"="

<< edge\_tag[i].weight

<< endl;

edge\_tag[i].visit = true;

++count\_vex;

}

if (count\_vex == Vexnum - 1) {

break;

}

++i;

}

## Data Normalization Notes

The theoretical results should time a constant c, which is:

c=average of experimental results/average of theoretical results

For this chart, c=105.3333333/82.66666667=1.274193548

## Output Numerical Data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n | 8 | 16 | 32 | avg |
| Theo | 24 | 64 | 160 | 82.66666667 |
| Exp | 89 | 102 | 125 | 105.3333333 |
| Adj Theo | 30.58064516 | 81.5483871 | 203.8709677 |  |
| log(n,2) | 3 | 4 | 5 |  |
| log(Exo,2) | 6.475733431 | 6.672425342 | 6.965784285 |  |
| log(Ad Th,2) | 4.934546939 | 6.349584438 | 7.671512533 |  |

## Graph

## Graph Observations

It can be seen from the graph that, even though the slope of theoretical results and experimental results are almost the same, the slope of O(n log n) is always growing faster compared to the other.

# Conclusions

Base on the graph, we can see the complexity of this algorithm is O(n log n).