CSC 4310/6310

Final Project Submission

Fall 2021

Team 3

Seung Lee, Wilburt Liu, Jay Alli, Vadim Kim

Prim’s algorithm is a greedy algorithm that finds the minimum spanning tree of a weighted, undirected graph. It iteratively adds a new vertex it to a set of vertices which represent the nodes in the MST. We explored the implementation of this algorithm in parallel using OpenMP in C++.

The program implements Prim’s algorithm based on user input. In short, the steps are explained below:

N = # of nodes that the user inputs

edge [N][N] = adjacency matrix

User inputs the weight values for the corresponding edge

**Initialize**: total distance begins at 0, store first node into set of vertices, delete node, increment number of nodes in the set

calculate for every node

set minimum distance to 1000

**for** every node

(execute in parallel)

if weight is 0 or more than minimum distance, skip

else

(execute single thread at a time)

update minimum distance to weight, set new element as the column

add local minimum distance to total

add next node to set and subtract elements of that column

increment number of nodes in set

print set of vertices (converted to letters) and total minimum distance

To make the output easier to understand and to better match conventional visual graph representations, the numbers of the nodes were converted to letters.

The graph from Slide 15, Lecture 18 (shown below) was used to test if the program correctly outputs the same MST

Diagram

Description automatically generated

The output matches the edges on the diagram.

If start with **a** as the root node, then the next node is **b**, and then **d**, then **c**, then **e**,

and then from **a** to **e** (Notice that the total minimum distance of 8 corresponds with the graph above.

Graphical user interface, text, application, email

Description automatically generated