CS6013: Advanced Data Structures and Algorithms

Programming Assignment III (out of 10 marks)
(Start Date: Saturday, 12 November 2022)
(Submission Deadline: 11:59 pm, Sunday, 20 November 2022)

Steiner Tree Problem

The assignment is about implementing the 2-factor approximation algorithm for the Steiner tree problem which was taught in the class (see Chapter 3 of 'Approximation Algorithms', by Vijay Vazirani, which has been uploaded under 'Resources/Course Outline' in Google Classroom). Assume $n \leq 15$. There is a file containing an $n \times n$ symmetric matrix, say A, whose every entry is a non-negative integer. We shall use $A_{i,j}$ to denote the i-jth entry of A. The matrix A is the weighted adjacency matrix of a graph G on n vertices. If $A_{i,j} > 0$, then it denotes the weight of the edge between the i-th vertex and the j-th vertex in G. If $A_{i,j} = 0$, it implies that there is no edge between the i-th and the j-th vertex. An example matrix A is given below for clarity.
(Matrix A)
0 3 0 9 1
3 0 5 0 2
0 5 0 3 7
9 0 3 0 1
1 2 7 1 0
The above matrix says (i) the weight of the edge between Vertex 1 and Vertex 4 is 9, (ii) the weight of the edge between Vertex 4 and Vertex 3 is 3, (iii) there is no edge between Vertex 2 and Vertex 4, etc..

The program reads matrix A from the file. It then asks the user to give the list of the Steiner vertices. The program then converts this into an instance of the Metric Steiner Tree problem, gives a 2-factor approximate solution to this instance, and then translates this solution into the original instance (as taught in class). Finally, the program outputs the 2-factor approximate Steiner tree.

Sample Output:
Sample:
The input matrix A the program read from the file is displayed below:
0 3 0 4 1
3 0 4 0 1
0 4 0 3 1
4 0 3 0 1
1 1 1 1 0
List all the Steiner vertices (type * to quit):
5
*
The 2-factor approximate tree we have computed is given below (we describe this tree by listing all the neighbors of all the vertices in the tree):
Neighbors of Vertex 1: 5
Neighbors of Vertex 2: 5
Neighbors of Vertex 3: 5
Neighbors of Vertex 4: 5
Neighbors of Vertex 5: 1 2 3 4

Program Related Instructions

1. You can write your program in one of C, C++, Java, or Python.

Submission Guidelines

1. Your submission will be one zip file named <roll-number>.zip , where you replace roll-number by your roll number (e.g. cs22mtech11003.zip), all in small letters. The compressed file should contain the below mentioned files:

   (a) Programming files (please do not submit python notebooks or IDE files). **The entire source code has to be in one file named main_prog.c (or main_prog.cpp, or ...).**

   (b) **No need to submit a report**. However, if you wish you may submit a text/doc file giving a detailed description of your program. No marks for this.

   (c) Upload your zip file in Google Classroom at Classwork→Programming Assignment 3. No delays permitted.

2. Failure to comply with instructions (file-naming, upload, input/output specifications) will result in your submission not being evaluated (and you being awarded 0 for the assignment).

3. **Plagiarism policy**: If we find a case of plagiarism in your assignment (i.e. copying of code, either from the internet, or from each other, in part or whole), you will be awarded a zero and will lead to a FR grade for the course in line with the department Plagiarism Policy (https://cse.iith.ac.in/academics/plagiarism-policy.html). Note that we will not distinguish between a person who has copied, or has allowed his/her code to be copied; both will be equally awarded a zero for the submission.

Evaluation Scheme

Your assignment will be awarded marks based on the following aspects:

- Code clarity (includes comments, indentation, naming of variables and functions, etc.): 1 mark.
- How perfect is the output: 4 marks.
- How well you explain the programming logic to the TA: 5 marks.