

Programming Assignment 1

Total Marks: 5

Spanning tree with colors Consider a graph G with weights on edges and each edge colored red or blue. Given a threshold T , we want to find a spanning tree whose weight is at most T and want to minimize the number of red edges. Formally, we want to find

$$\min\{k : \text{there is a spanning tree with } k \text{ red edges and weight at most } T\}.$$

Output both the number of red edges and the weight of the optimal tree (which will be at most T). You can assume that the input graph has a spanning tree with weight at most T .

See input/output files for some small examples.

Instructions

No. of the lines in the Input = 3 + number of edges. The $(i + 3)$ -th line gives the endpoints of i -th edge u_i, v_i , its weight w_i and its color r_i (1 if red, 0 otherwise).

Line 1: n (the number of vertices)

Line 2: m (the number of edges)

Line 3: T (the weight threshold)

\vdots

Line $i + 3$: $u_i v_i w_i r_i$

Output :

Line 1: number of red edges in the optimal tree

Line 2: weight of the optimal tree

- Programming Language: C++. We will compile your code with g++. Make sure that it works.
- Submission: put your code in a file named XXX.cpp where XXX is your roll number. Also, write a short explanation (a paragraph) of what your algorithm does, put this in XXX.pdf. Also mention in the pdf compilation options that need to be used. The two files should be uploaded on Moodle (do not zip/compress).
- Given files: In the `colorMST` folder, you will find: (i) `helper.cpp` (a c++ code showing expected input/output, feel free to use) (ii) Few sample input and output files, (iii) two executable files, which can be used to get the correct output on any input.
- Running time: we will test your code on some similar size instances as given in the sample input files (few of small size, few of large size).
- Academic integrity: Mention all references if you have referred to any resources while working on this assignment in the pdf. You are supposed to do the assignment on your own and not discuss with anyone else. We will do a plagiarism check on your submission using MOSS. It's fairly sophisticated and can detect even when you have made modifications in someone else's code. Any cases found with significant overlap will be sent to DADAC. If DADAC finds it to be a case of plagiarism, then the penalty is zero in the assignment and final course grade reduced by 1 point.
- Grading: We will use mars.cse for testing, with a timeout of 5 seconds for each input. The test inputs will be of varying sizes. Total marks will be equally distributed for the test inputs.