CS218 Design and analysis of algorithms **Jan-Apr 2025**

Programming Assignment 2

Total Marks: 5

Note: You are not supposed to directly use any linear programming (LP) or integer linear programming solver.

You can directly use any code solving network flow. Edmonds-Karp algorithm will be faster than Ford-Fulkerson.

There is am FMCG company, which produces multiple products. It needs to decide its sales budget for various products across various cities. For each pair of a product and a city, they are given a range (a lower bound and an upper bound) on the budget. They are also given a ranges for the total budget of any city and also for the total budget of any product. We need to figure out if there is a budget allocation which respects all the ranges.

Abstractly, you have to fill in an $m \times n$ matrix X with integers, while satisfying the below constraints. Let $x_{i,j}$ be the (i, j) entry of the matrix.

- $\ell_{i,j} \leq x_{i,j} \leq u_{i,j}$ for every $1 \leq i \leq m$, $1 \leq j \leq n$.
- $r_i \leq x_{i,1} + x_{i,2} + \cdots + x_{i,n} \leq R_i$ for every $1 \leq i \leq m$.
- $c_j \leq x_{1,j} + x_{2,j} + \cdots + x_{m,j} \leq C_j$ for every $1 \leq j \leq n$.

Here $\{\ell_{i,j}, u_{i,j}\}, \{r_i, R_i\}, \{c_j, C_j\}$ are given as input. You have to output

- whether it is possible to fill in the matrix (output 1 if possible otherwise 0).
- if possible what is the maximum possible total budget $\sum_{i,j} x_{i,j}$.
- if possible what is the minimum possible total budget $\sum_{i,j} x_{i,j}$.

Example input: $m = 3, n = 3$.

$$[\ell_{i,j}] = \begin{pmatrix} 1800 & 1000 & 500 \\ 1500 & 700 & 500 \\ 1400 & 800 & 600 \end{pmatrix}$$

$$[u_{i,j}] = \begin{pmatrix} 1900 & 1100 & 600 \\ 1600 & 700 & 600 \\ 1500 & 900 & 700 \end{pmatrix}$$

$$(r_1, R_1, r_2, R_2, r_3, R_3) = (3480, 3490, 2835, 2840, 2960, 3000)$$

$$(c_1, C_1, c_2, C_2, c_3, C_3) = (4870, 4895, 2600, 2605, 1805, 1815)$$

Output:
1
9315
9275

In the above example, we can fill in the matrix as follows to maximize the total budget (not expected in the output).

$$\begin{pmatrix} 1900 & 1020 & 570 \\ 1595 & 700 & 545 \\ 1400 & 885 & 700 \end{pmatrix}$$

We can fill in the matrix as follows to minimize the total budget.

$$\begin{pmatrix} 1900 & 1040 & 540 \\ 1570 & 700 & 565 \\ 1400 & 860 & 700 \end{pmatrix}$$

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: m (the number of rows)
Line 2: n (the number of columns)
Line 3: $\ell_{1,1}\ \ell_{1,2} \cdots \ell_{1,n}$
$\vdots$
Line 2+m: $\ell_{m,1}\ \ell_{m,2} \cdots \ell_{m,n}$
Line 3+m: $u_{1,1}\ u_{1,2} \cdots u_{1,n}$
$\vdots$
Line 2+2m: $u_{m,1}\ u_{m,2} \cdots u_{m,n}$
Line 3+2m: $r_1\ R_1 \cdots r_m\ R_m$
Line 4+2m: $c_1\ C_1 \cdots c_n\ C_n$
Output :
Line 1: 1 or 0 (1 if it is possible to fill in the matrix)
Line 2: minimum possible total budget
Line 3: maximum possible total budget

- 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 `fillMatrix` 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.
- 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.