CS-2580: Hw2 Supply Chain

Final Due Date: Sunday, February 27, 11:59pm

1 Problem Statement

A major humanitarian crisis has just occurred and the US government is scrambling to mobilize their resources and provide aid ASAP. Their key problem is how to effectively utilize the global supply chain network (i.e. trucks, trains, planes, ships, ...) to move the goods. The congress quickly realizes this is a tricky problem and come to YOU (the latest hot-shot humanitarian logistics startup, which just happens to be you and your partner working out of your parents basement) for assistance.

The input data for the Supply Chain problem consists of a undirected graph $G = \langle V, E \rangle$ and three properties $s(v), d(v), w(e)$. For each vertex, $v$, $s(v)$ indicates how many supplies are stored and $d(v)$ indicates how many supplies are needed at that vertex. For each edge, $e = \langle u, v \rangle$, $w(e)$ indicates the maximum amount of supplies that can be moved directly from vertex $u$ to vertex $v$. The optimization problem is to move as many units as possible from the supply points to the demand points while adhering to the edge capacities.

2 Assignment

Write a algorithm to solve the Supply Chain problem, you can apply any technique you prefer, including but not limited to, LS, CP, IP, LP, DP, brute force etc. Your algorithm should be able to perform on all of the data sets in the course directory (the include supply chains with as many as 10,000 vertices). The assignment should be performed by teams of at most 2 people. We always expect

- both source files and binary programs, if any, of the working algorithm;
- a specification on how to run the program;
- a brief report in plain text containing, the names of each team member, a brief discussion of your solution strategy, implementation techniques and experimental observations. The report should be concise. An example report can be found here,

/course/cs258/data/README_example

3 I/O Specification

**Format** The input consists of $|V| + |E| + 3$ lines. The first line contains one number, $|V|$. It is followed by $|V|$ lines, where each line $i$ represents $s(i), d(i)$. At the end of this list there is a blank line followed by one number, $|E|$. It is followed by $|E|$ lines, where each line $i$ represents an edge $u(i), v(i), w(i)$ where $0 \leq u_i, v_i < |V|$.
[Input Format]
|V|
s_0 d_0
s_1 d_1
...
|E|
u_0 v_0 w_0
u_1 v_1 w_1
...

The output has two lines. The first line contains two numbers: the first number is the best objective value found by the algorithm; the second number is a flag, 1 if the algorithm is able to prove the optimality, otherwise 0. The second line is the amount of supplies moved on each edge. It should be a positive value if the supplies move from \( u_i \) to \( v_i \) and a negative value if they move from \( v_i \) to \( u_i \).

[Output Format]
obj opt_flag
m(0) m(1) ... m(|E|-1)

Example  For an example input see
/course/cs258/data/supply/sc_9_1

For an example output see
/course/cs258/data/supply/sc_9_1.out

Instructions  We will run your submission using the command: 
./sc <timelimit> <filename>

For example: 
./sc 60 /course/cs258/data/supply/sc_9_1
means the program will use sc_9_1 as input and will run at most 60 seconds.

We use stdout for output. Output to other stream will be ignored. Your submission will be tested on department’s linux machine. If your algorithm is a standalone program, please name it sc, otherwise, please specify the compilation procedure and provide a script that follows the above format to run the program.

Resources  Some sample input files are available in: /course/cs258/data/supply

4 Remarks

Handin  Command: /course/cs258/bin/cs258_handin hw2
All of the files in the current directory and sub-directories will be submitted. Only the last submission will be marked.

Questions  Please contact the class GTA Carleton (cjc@cs.brown.edu).
**Time Limit**  You can assume that the time limit will be something between 10 and 600 seconds.

**Hint**  An optimal value for $sc_{10000.3}$ is 423926.

**Warning**  This is a first time assignment, be prepared for clarifications and bug-fixes.