Summer term 2019
Dr. U. Brenner

Research Institute for Discrete Mathematics
University of Bonn

Linear and Integer Optimization
Programming Exercise 2

Implement the Network Simplex Algorithm for solving the Minimum-Cost Flow Problem. We restrict ourselves to simple graphs and only consider positive edge costs. You may assume that the given instance is feasible.
The program has to be implemented in C/C++ and compile with either the GNU compilers gcc/g++ or clang/clang++. The program should be run from the command line and read in a text file, whose name is given as an argument.
You may use (a modified version of) the class Graph from the lecture course "Algorithmische Mathematik I" in the winter term 2018/2019. All data structures presented in this lecture course can be found here: hier:

`http://www.or.uni-bonn.de/~hougardy/alma/alma.html`

You can use all these programs and data structures. Moreover, you can use the STL, but no other external libraries.

Reading the input data:
The format of an input file is as follows:

```
Number of nodes
Supply0
Supply1
...
Number of edges
Node0a Node0b Capacity0 Cost0
Node1a Node1b Capacity1 Cost1
...
```

All entries are integers, and you may assume that the sum of all absolute values of the entries is smaller than 2^{31}.
The first line contains a single positive number n denoting the number of nodes. The following n lines specify the b values of the nodes, so each of these lines contains exactly one number. We assume that the nodes are labeled from 0 to $n-1$. The number in line i denotes the b value of node $i-2$ (for $i = 2, \ldots, n+1$).
The next line (after these $n+1$ lines) denotes the number of edges. After that, each of the following lines specifies exactly one edge. The first two entries are two non-negative numbers encoding the two end-nodes of the edge (where the edge is directed from the first node towards the second node). The third number in the line is a positive number denoting the capacity of the edge. The fourth (and last) entry is again a positive number denoting the cost of the edge. The index of any edge is given by the line number in the input file: line i encodes the edge with index $i-3-n$ (for $i = n+3, \ldots, m+n+2$, where m is the number of edges).

Output format: In the first line, the program has to write the cost of the computed solution. Any of the following lines contains exactly two numbers. The first number is the index of an edge, and the second number is the flow value on this edge. The edge indices have to be sorted in ascending order. Only edges with positive flow value have to written to the output.

Example: An input file encoding a graph with five nodes and seven edges can look like this:

```
5
2
0
0
3
-5
7
0 1 1 1
1 2 1 1
0 2 5 3
0 3 7 5
3 2 3 4
2 4 2 1
3 4 3 2
```

The output has to look like this:

```
13
0 1
1 1
2 1
5 2
6 3
```

You can find test instances for your program on the web page of the exercises:

`http://www.or.uni-bonn.de/lectures/ss19/lgo_uebung_ss19.html`

(20 points)

Due date: Thursday, June 6, 2019, before the lecture. Please send your solution via e-mail to you tutor.