Research Institute for Discrete Mathematics
Chip Design
Summer term 2015

Prof. Dr. Jens Vygen
Markus Ahrens, M. Sc.

Programming Exercise 3

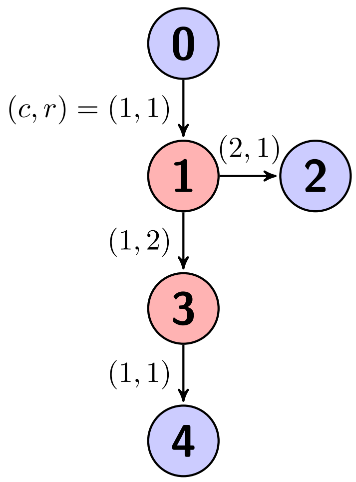
Implement the DYNAMIC PROGRAMMING BUFFERING ALGORITHM from the lecture. The running time must be $\mathcal{O}(|L|^2|V(A)|^2)$ and the program call must be

PROGRAM <INPUTFILE> <OUTPUTFILE>

The source code must be written in C or C++ and compile with GCC on Linux. It should be well documented [1]. You are allowed to use standard headers including the STL, but no other external libraries.

Input: The input file is a text file containing the number of vertices $n := |V(A)|$ of the arborescence A as its first line. The set of vertices is $\{0, \dots, n-1\}$. Each of the following $n-1$ lines contains four natural numbers $v\ w\ c\ r$, encoding an edge (v, w) with capacitance c and resistance r.
The next line contains the number k of sinks in $V(A)$ and the resistance of the root vertex. Each of the following k lines contains three natural numbers $v\ c_v\ rat_v$, encoding that v is a sink with required arrival time rat_v and that the pin placed at v has input capacitance c_v.
The next line contains the number of buffers $|L|$ in the library L. Each of the following $|L|$ lines contains two natural numbers $c\ r$, where c is the input capacitance of the buffer and r is its resistance. The buffer encoded in the l-th such line has index l. The delay of a circuit (buffer or circuit at the root) with resistance r is $r \cdot dc$, where dc is the downstream capacitance. Buffers can be placed at vertices that are not a sink and not the root.
An example input file looks like this (buffers can only be placed at red vertices):

```
5
0 1 1 1
1 2 2 1
1 3 1 2
3 4 1 1
2 2
2 1 18
4 1 19
1
2 1
```



[1] This can be achieved by using comments and – much more importantly – self-documenting code.

Output:

The task is to find a buffering that maximizes the worst slack. The output file must start with a line containing this slack and the number of buffers b in your solution. This should then be followed by b lines, encoding the positions of the buffers. A line v l means that buffer l is used at vertex v.

In the following example, a buffer of type 1 is placed at vertex 1.

```
-2 1
1 1
```

Test instances will be provided on the website of the exercise class

http://www.or.uni-bonn.de/lectures/ss15/chipss15_ex.html.

The complete source code must be sent to *ahrens@or.uni-bonn.de* until

Thursday, June 25, 12:15h.

(20 points)

In case of any questions feel free to contact me at ahrens@or.uni-bonn.de .