Problem 3: Waterfront

(Proposed by Eugen Nodea. We thank Tamio-Vesa Nakajima for this editorial.)

We will describe the solution in several stages, starting from a brute-force solution, and then gradually optimising it.

Brute force solution. A simple brute-force solution is the following:

- Find the tree which will be the tallest *at the end of the M days*.
- Find the first moment at which this tree can be cut, if such a moment exists.
- Cut the tree at that moment, if it exists.
- If not, output it's height.

To see (broadly) why this is correct, suppose that we try to see if the solution can be at most S. Then, based on the heights at the end of the M days, we know how many times each tree must be cut (according to the formula $\lceil S - \mathit{finalHeight}/x \rceil$). Each cut can then be done in some suffix of days. We then need to assign cuts to days in some way. This is an example of an *activity selection* problem. We have M days in which we can do k activities daily, and each activity can be done in some suffix of days. It is well known that such problems can be solved using a greedy approach – which is exactly what we do.

The previous remarks justify our brute force algorithm. If the real solution is S, then we note that the set of tree cuts done by our brute force algorithm are precisely the tree cuts necessary for solution all solutions greater or equal to S in decreasing order – thus stopping at precisely the correct solution. Furthermore, we can see that our greedy solution is correct, since the set of assigned activities will be the same as if we only considered those cuts (even if certain cuts may be done in different days).

Optimisation 1. First we optimise the part of the solution where we find the first point at which a cut can be made. Note that it is easy to check using arithmetic and some bookkeeping the first point in time in which a certain tree is tall enough to be cut. We now need to find the first day in which it can be cut. If we model the days as an array $v[1], \ldots, v[M]$, where $v[i]$ is the number of cuts allowed in day i, then we want to find the first non-zero value in some suffix $v[a], \ldots, v[M]$. Then we want to decrement that value (that is where the cut is done).

How do we do this? Suppose we consider a partition of indices $1, \ldots, M$ where all adjacent indices i where $v[i] = 0$ are joined into the same partition. In this case to find the first nonzero value to the right of index a, we find the set to which a belongs, we find the rightmost index r in this partition, and:

- if $v[r] = 0$, then by the definition of the partition the next nonzero index if $r + 1$;
- otherwise $r = a$ and a itself can be decremented.

This partition can be efficiently maintained in $\log^* M$ complexity using a disjoint set data structure.

Optimisation 2. We now optimise the way of choosing the tree to cut. We previously said that we always choose the tree that is tallest at the end of the M days. In essence this can be simulated using a priority queue. The keys are the heights at the end of the M days, and the values are the indices of the trees.

Optimisation 3. The idea from the previous paragraph can be optimised further. Note that we actually have the following operations on our priority queue:

- Finding the maximum key-value pair.
- Decrementing the maximum key by a constant, x.

This type of priority queue can be implemented in constant time for each operation, with $n \log n$ precalculation time. To do this, maintain two data structures, a stack S and a queue Q. Insert all the key-value pairs into S in increasing order (so that the maximum value is at the top of the stack). When trying to get the maximum key-value pair, take the maximum of the key-value pair from among the top of S and the front of Q. To decrement it's key, remove it from S or Q respectively, and add it into Q (at the back), with a decremented key. We leave the proof of correctness as an exercise.

Scientific committee

The problems were proposed and prepared by:

- Adrian Panaete (chair) - "A.T. Laurian" National College, Botoșani
- Ionel-Vasile Piț-Rada - "Traian" National College, Drobeta Turnu Severin
- Maria-Alexa Tudose - University of Oxford, UK
- Gheorghe-Eugen Nodea - "Tudor Vladimirescu" National College, Târgu Jiu
- Tamio-Vesa Nakajima - Oxford, Computer Science department, UK
- Mihai Bunget - "Tudor Vladimirescu" National College, Târgu Jiu
- Andrei-Costin Constantinescu - University of Oxford, UK
- Ciprian-Daniel Chește - "Grigore C. Moisil" Technological High School, Buzău
- Lucian Bicsi - University of Bucharest
- Dan Pracsiu - "Emil Racoviță", Theoretical High School, Vaslui
- Ioan-Cristian Pop - Polytechnical University, Bucharest
- Theodor-Gabriel Tulbă-Lecu - Polytechnical University, Bucharest
- Raluca-Veronica Costineanu - "Ștefan cel Mare" National College, Suceava