Job Lookup

Problem author: Vitaly Aksenov; problem developer: Mikhail Dvorkin

This problem can be solved by dynamic programming “over subsegments”.

Consider a subproblem for a segment of people $[i, j]$. We want to arrange them as a subtree of the global hierarchy tree in the optimal way. Let's denote a_{ij} to be the minimal possible communication cost induced by the edges in this subtree. That is, each communication path that costs $c_{uv} \cdot d_{uv}$ can be fragmented as the sum of d_{uv} payments of size c_{uv} being paid in each edge of the path. So in our dynamic programming value a_{ij} we will only consider the part of communication cost that is paid in the edges of the constructed subtree.

To calculate a_{ij} simply iterate over all possible root candidates $k \in [i, j]$. For a specific k the communication cost in the subtree $[i, j]$ is composed of: $a_{i,k-1}$, $a_{k+1,j}$, the communication cost paid in the edge from k to its left child (if any), and the communication cost paid in the edge from k to its right child (if any).

The communication cost paid in the edge from k to its left child is the sum of c_{uv} over all pairs (u, v) such that u is in $[i, k-1]$ and v is not (indeed, these are all pairs of people that do pay in this edge). This value is simply a sum of two subrectangles in the matrix c. The same obviously applies to the cost in the edge to the right child.

If the prefix sums (or a similar data structure) is precalculated for the matrix c, the cost for specific k can be calculated in $O(1)$ time, the value of a_{ij} — in linear time, and the entire problem — in cubic time.

Some information about the origin and the relevance of the problem can be found in the SplayNet paper, section III A: `https://www.univie.ac.at/ct/stefan/ton15splay.pdf`