Analysis

Fountain

2020 - Tbilisi - Georgia

We will describe how to solve the last subtask which was enough to achieve a full score.

Subtask 3:

The solution consists of three parts: building the graph (which is a tree), preprocessing the tree and answering the queries.

- For each reservoir we determine the next one (i.e. where the water flows into when the given reservoir overflows). To do this efficiently, we can go from bottom to top and maintain the stack of the reservoirs in ascending order of their diameters. Each time a new reservoir is processed we remove smaller or equal reservoirs from the top of the stack and insert the current reservoir, determining the next reservoir along the way. This can be done in O(N). After that we connect each reservoir to the next one and we end up with a directed tree with a root in node 0 (that corresponds to the waterways).
- We use the method called “binary lifting”, most commonly used in LCA algorithm, where for each node we determine its 1st ancestor, 2nd ancestor, 4th ancestor, …, 2^k-th ancestor. Additionally we precompute the total amount of water that the reservoirs up to the corresponding ancestor can hold. Since k is of the order of logN, we will need O(N logN) time and O(N logN) memory for this part.
- For each query we keep moving to the farthest precomputed ancestor while the given amount of water doesn’t exceed the sum of the capacities of the nodes traversed and find the answer. This step has O(Q logN) complexity.