KONZERVE

To start with, let us calculate the number of points we would initially get for a shot at height h. Let the sequences B(i), G(i) and W(i) be the number of black, grey and white cans in column i. Let $X_i(h)$ be the value of the can in column i at height h, and X(h) the total points won if our first bullet is fired at height h. Obviously,

$$X(h) = \sum_{i=1}^{N} X_i(h)$$

The above relation does not enable us to efficiently calculate X(h), so we introduce auxiliary sequences $Y_i(h)$, the difference in value of cans at heights h and h−1 in column i, useful because they can have at most four distinct values other than zero:

$$Y_i(h) = X_i(h) - X_i(h-1)$$

$$X_i(h) = \sum_{i=1}^{h} Y_i(h)$$

Let Y(h) be the sum of all sequences $Y_i(h)$. Y(h) can be efficiently calculated if we know the positions in $Y_i(h)$ containing values other than zero. These positions are 1, B(i)+1, B(i)+G(i)+1 and B(i)+G(i)+W(i)+1.

Finally, we obtain X(h) by summing the sequence Y(h):

$$X(h) = \sum_{i=1}^{N} X_i(h) = \sum_{i=1}^{N}\sum_{j=1}^{h} Y_i(h) = \sum_{j=1}^{h}\sum_{i=1}^{N} Y_i(h) = \sum_{j=1}^{h} Y(h) = X(h-1) + Y(h)$$

Now that we know the values of cans at every height, we need to determine, for every bullet, which set of cans it hits. One of the way to determine this is to insert all heights from 1 to 3300000 into a data structure that supports efficient removal of elements and can answer queries of the form "What is the K-th smallest number in the structure?". One such data structure is a complete binary tree.

The leaves of the tree contain heights from 1 to 3300000, and each internal node maintains a count of active (not erased) heights in its subtree. That infromation is enough to efficiently find the K-th smallest element and to erase it.

PUTEVI

First let us orient the tree, so that for each node, we know its parent node is and its children nodes are.

Now choose a vertex v and observe all paths going through v and is entirely in its subtree (so that v is the "highest" vertex in the path). Such paths can be divided split into two categories:

1. Paths that end in vertex v – let their total weight be A(v);
2. Paths that go through v, but do not end in it – let their total weight be B(v).

Let C be the set of children of vertex v. Let w_u be the weight of the edge between v and its child u.

It should be obvious that:

$$A(v) = \sum_{u \in D} (A(u) + 1) \cdot c_u \ .$$

Also, we can calculate B(v) as:

$$B(v) = \sum_{\substack{u1<u2 \\ u1,u2 \in D}} (A(u1)+1) \cdot c_{u1} \cdot (A(u2)+1) \cdot c_{u2} = \frac{1}{2}\left(\left(\sum_{u \in D} (A(u)+1) \cdot c_u \right)^2 - \sum_{u \in D} (A(u)+1)^2 \cdot {c_u}^2 \right).$$

For B(v) we use the right side of the equation because we can calculate it in linear time.

The solution is the sum of A(v) and B(v) for all vertices in the tree.