Heroes of Might

Problem author: Nikolay Budin; problem developer: Borys Minaiev

The full editorial of this problem could be pretty long, so we just provide some key ideas required for the solution without going into much detail.

Let's first discuss the slow solution, which works when the total number of rounds is not big. For each group of peasants, we can generate an array of integers k_i — number of peasants killed on i-th attack of this group. For example if damage is equal to 15, the health of each peasant is 10, and there is a group of 4 peasants, the corresponding array is equal to $[1, 2, 1]$.

After constructing an array for each group, we can "merge" them into one big array in a way which preserves the initial ordering of elements in each array. Such a merged array corresponds to some ordering in which dragon attacks groups. Optimal merging should maximize the sum of all prefix sums of the generated array.

How to merge arrays? Let's split each array into several continuous intervals, and for each of them calculate a_i — sum of elements in the segment divided by the length of the segment. Over all possible splits we choose one with special properties:

- $a_1 > a_2 > ... > a_n$
- $(a_1, a_2, ..., a_n)$ — lexicographically largest

Such a split could be computed greedily. First, find the prefix with largest a_1, cut it, recursively find other prefixes.

There is also a nice geometrical interpretation of such a split. We can draw points $(i, \sum_{j \le i} k_j)$, and find the upper convex hull of them. Each segment of the convex hull corresponds to the segment in the optimal array split.

It could be proven that each segment of the split could stay continuous inside optimally merged arrays. Moreover, to determine the order in which segments for different groups should be merged, we can just sort them by a_i.

It also could be proven that the optimal split has $O(log)$ segments.

The only question left is how to build such segments for larger constraints. We need to find a convex hull of points $(i, \lfloor \frac{d \cdot i}{h_p} \rfloor)$. There are different possible approaches. One of the easiest is to use continued fractions. There is a detailed description of the algorithm for finding the convex hull of lattice points under the line in https://cp-algorithms.com/algebra/continued-fractions.html.

We also need to handle the last point carefully as it doesn't follow formula $\lfloor \frac{d \cdot i}{h_p} \rfloor$.