F. GukiZ Height

Editorial

In this problem, you are given a array $A[0, \ldots, N-1]$ and integer H. Initially, you start with a value $V = 0$. You also know, that at time $i = 1, 2, \ldots$ V changes by $A_{(i-1)\ mod\ N}$ and H decreases by i. The goal is to calculate the minimum time t at which $V \geq H$.

First, let's reformulate the problem a little bit. Rather than decrease H by i at time i, we can add i to V then. This allows us to leave H fixed for the whole process. Now, we have to deal with the following problem: given a function f, with value $f(0) = 0$ and $f(i) = f(i-1) + A_{(i-1)\ mod\ N} + i$ for $i > 0$, find the smallest t, such that $f(t) \geq H$.

First, you can notice that based on the above definition of f, you can easily compute its value at any given time i. It looks like we can use ternary search here, but its not so straightforward - notice that f may not be convex on its whole domain.

Fortunately, we can decouple f into N new functions. Let $f_k(i) := f(i \cdot N + k)$ for $0 \leq k < N$. In other words, f_k is f defined only at points j for which $j\ mod\ N = k$. Here comes the crucial observation: each f_k is convex. This is true, because the difference between its two consecutive values first reduces for some time and then becomes larger and larger. Based on this observation, we can apply ternary search separately for each f_k to find the first point for which $f_k \geq H$. In order to find the final result, we only need to return the smallest of points computed for each f_k.

Another, more clever approach allows to avoid ternary search here. First, we can check if there is any point $i < N$ such that $f(i) \geq H$. If we are lucky and find any such point, we can return the smallest such one. Otherwise, we know that for any $i < N$, $f(i) < H$, thus we know that for each function f_k, there is only one point $i \geq 0$, for which $f_k(i)$ is smaller than H and $f_k(i+1)$ is greater or equal than H. This allows us to use binary search instead of ternary search here.

Last but not least, implementation details. Notice, that we can use binary search over a range $[0, 2 \cdot 10^9]$, because for a given constraints, $f(2 \cdot 10^9)$ is always greater or equal than H. You should also pay attention to overflows calculating a middle point of a range.