Algorithm Design Laboratory with Applications

Prof. Stefano Leucci

Problem: *Lazy Hikers.*

A group of lazy hikers decides to summit the Great Pebble, a massive mountain that is $H \in \mathbb{N}^+$ meters tall, by following a path from the base to the top. The path is $D \in \mathbb{N}^+$ meters long and starts from an elevation of exactly 0 meters above sea level.

Along the path there are $n \in \mathbb{N}^+$ refuges where the group can stop and rest for the night. The i-th refuge is encountered $d_i \in \mathbb{N}^+$ meters after the beginning of the path (with $0 < d < D$) and has an elevation of $h_i \in \mathbb{N}^+$ meters (with $0 < h_i < H$). The elevation of the refuges is monotonically non-decreasing (w.r.t. the order in which the refuges are encountered on the path).

The hikers, being lazy, have some specific constraints about their journey: they do not want to walk more than W meters per day, they do not want to climb (i.e., increase their elevation by) more than C meters per day, and they absolutely do not want to sleep outside of a refuge.

Design an algorithm that, given n, H, D, and all the values d_i and h_i, computes the minimum number η of days needed for the hikes to reach the summit of the Great Pebble.

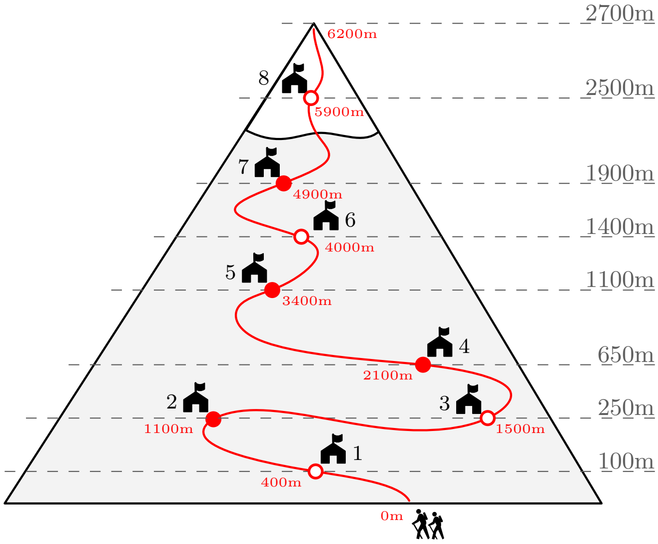
Input. The input consists of a set of instances, or *test-cases*, of the previous problem. The first line contains the number T of test-cases. The first line of each test case contains the integers n, H, D, C, and W. The next line contains the n integers $h_1, \dots, h_n$. The third and final line of the test case contains the n integers $d_1, \dots, d_n$.

Output. The output consists of T lines. The i-th line is the answer to the i-th test-case and contains the integer η.

Assumptions. $1 \le T \le 10$; $1 \le n \le 2^{20}$; $1 \le H, D, W, C < 2^{31}$;
It is always possible to reach the summit while satisfying all the hikers' constraints.

Example.

The following example shows an instance with $n = 8$, $H = 2700$, $D = 6200$. The distances in red indicate the length d_i of the segment between the start of the path and the i-th refuge (for $i = 1, \dots, n$), except for the distance at the summit which is the total length of the path. An optimal solution for $C = 800$ and $W = 2000$ is $\eta = 5$. A possible schedule that allows to reach the mountain top in 5 days stops at refuges 2, 4, 5, and 7 (depicted with a solid red dot).



Input (corresponding to the above picture):

```
1
8 2700 6200 800 2000
100 250 250 650 1100 1400 1900 2500
400 1100 1500 2100 3400 4000 4900 5900
```

Output:

```
5
```

Requirements. Your algorithm should require $O(n)$ time (with reasonable hidden constants).

Notes. A reasonable implementation should not require more than 0.5 seconds for each input file.