ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Algorithms Lab HS22
Department of Computer Science
Prof. Dr. A. Steger, Prof. Dr. E. Welzl
cadmo.ethz.ch/education/lectures/HS22/algolab

Exercise – *Astérix the Gaul*

After a long struggle all of Gaul is occupied by the Romans—all of it, except for a tiny village that is still holding out. After months of siege Centurion Caius Bonus manages to infiltrate the village. A spy discovers the secret that makes these Gauls invincible: a druid called Panoramix periodically prepares a magic potion. Upon receiving this information the Centurion orders to have the druid captured and interrogated for the recipe. Fortunately, the protagonist Astérix learns about the plan and now wants to warn Panoramix before the Romans reach him.

Panoramix is at distance D from Astérix and the Romans need T seconds in order to reach Panoramix. However, Astérix cannot run as he wishes—any suspicious movement will alert the Romans and he might end up imprisoned himself before he can warn Panoramix.

Astérix has a list M of n possible movements that will not raise any suspicion from the Romans. A movement consists of a traversed distance d and a number of seconds t that it takes to perform. Several movements in M may have the same parameters d and t, and you may think of M as a multiset. Astérix may perform each movement in M at most once, that is, the list of movements performed must form a submultiset of M.

Luckily, Astérix is equipped with a bottle of magic potion. Before starting his quest, he can take up to m gulps from the bottle. Taking i gulps of the potion allows Astérix to cover an additional distance of s_i in *every* movement he performs.

In order to reach Panoramix in time, Astérix must cover a distance of at least D in strictly less than T seconds. It is not required that he stops after covering a distance of exactly D, that is, just running past Panoramix is enough to warn him. Naturally, Astérix wants to minimize the amount of potion used because every bit of it is needed to fight those pesky Romans...

Input The first line of the input contains the number $t \leqslant 30$ of test cases. Each of the t test cases is described as follows.

- It starts with a line that contains four integers n m D T, separated by a space. They denote

  - n, the number of movements Astérix may perform ($1 \leqslant n \leqslant 30$);

  - m, the amount of magic potion Astérix has available ($0 \leqslant m \leqslant 10^5$);

  - D, Astérix's distance to Panoramix ($1 \leqslant D < 2^{50}$);

  - T, the number of seconds it takes for the Romans to reach the druid ($1 \leqslant T < 2^{50}$).

- The following n lines define the list M of movements Astérix may perform. Each line contains two integers d t, separated by a space, that denote the traversed distance ($1 \leqslant d < 2^{50}$) and the amount of time required for the movement ($1 \leqslant t < 2^{50}$).

- The following line contains m integers $s_1 \ldots s_m$, separated by a space, where s_i denotes the additional distance covered by every movement after taking i gulps of the magic potion initially ($1 \leqslant s_1 \leqslant s_2 \leqslant \ldots \leqslant s_m < 2^{50}$).

pseudo poly solution is not possible here

Output For each test case output a single line with a single integer that denotes the minimum number of gulps of magic potion Astérix needs to drink so as to warn Panoramix in time. If the available amount of potion does not suffice, output "Panoramix captured" instead

Points There are five groups of test sets, worth 100 points in total.

1. For the first group of test sets, worth 20 points, you may assume that $m = 0$ and $n \leqslant 20$.

2. For the second group of test sets, worth 20 points, you may assume that $m \leqslant 20$ and $n \leqslant 20$.

3. For the third group of test sets, worth 20 points, you may assume that $n \leqslant 20$.

4. For the fourth group of test sets, worth 40 points, there are no additional assumptions.

Corresponding sample test sets are contained in `testi.in/out`, for $i \in \{1, 2, 3, 4\}$.

Sample Input
```
3
2 0 10 3
9 1
2 2

2 3 13 7
3 3
5 3
1 2 3
3 3 21 12
20 6
17 5
3 10
5 7 8
```

Sample Output
```
Panoramix captured
3
0
```