Coding Assignment 3

Due: 21 November, 2025 6pm PT

About This Assignment

- Please download the file `CA3.zip` from Canvas (see the corresponding Canvas assignment). It contains all the visible test cases for the problem, as well as template code that handles input/output for you.
- Please submit your solution on Gradescope. Your solution will be evaluated on the visible test cases and a set of hidden test cases.

Radio Towers Again

Problem Description

Note: This coding assignment includes a subtask that admits much simpler solutions. Therefore, even if you cannot solve the full problem, you may still solve the subtask to get partial credit. See the subtasks section for information.

In LineLand, there are n cities located along the x axis with the i^{th} city located at $(A_i, 0)$.

In LineLand, a telecommunications company that manufactures radio towers. Recently, there was a tsunami that destroyed all radio towers in the region, and this company is now in charge of reconstructing the radio towers.

The company has decided to build exactly k radio towers in LineLand. Their radio towers demand an ample power source, and therefore **they can only be built in cities**. The towers are designed to *supposedly* have the same coverage range d. However, the actual coverage of a tower may be influenced by nearby buildings and terrain. After investigation, the company finds that a tower will only have p_i percent of its expected coverage when built in the i^{th} city. That is, if a tower is built in the i^{th} city, it can provide services to all cities j such that $|A_i - A_j| \le d \cdot p_i/100$.

The company is still trying to figure out the best coverage radius d, because a larger coverage radius would imply a higher cost for building the radio towers. Hence, they want to choose the smallest possible coverage radius such that it is possible to provide services to all cities. More specifically, they want to find the smallest d satisfying the following:

- There exist k cities such that all cities will be covered if we build a radio tower on each of the k cities.

Hints

- For a fixed value of d, can you figure out a fast way to determine if k radio towers suffice?
- Once you solve the above decision problem, how can you solve the original optimization problem?

Input

Each input file contains several test instances. The first line contains a single integer C, representing the number of test instances. The description of the C instances follows.

Each test instance consists of three lines:

- The first line contains two space-separated integers n, k.
- The second line contains n space-separated integers $p_1, p_2, \ldots, p_n$, where $p_i \in \{20, 40, 60, 80, 100\}$ for $i = 1, 2, \ldots, n$. (That is, the percentages can only be multiples of 20.)
- The third line contains n space-separated integers $A_1, A_2, \ldots, A_n$ denoting the location of the cities in LineLand.

- It is guaranteed that the city locations are given in increasing order, i.e., $A_i > A_{i-1}$ holds for every $1 < i \leq n$.

Output

For each test instance, your program should output the minimum coverage radius d times 60 (that is, $60d$). Under the above constraints, it can be proven that $60d$ must be an integer for the minimum possible value of d. In other words, the minimum possible value of d can be represented as a fraction $q/60$ for some integer q, and your program should output $q = 60d$.

Constraints

Each input file contains at most 2000 test instances. In a single input file, the total number of cities across all test instances is at most 10^5. The time limit (for all test cases in a single file) is 4 second for C/C++ and 10 seconds for Python.

Each test instance satisfies the following additional constraints:

- $1 \leq k \leq n \leq 10^5$.
- For $i = 1, 2, \ldots, n$, $p_i \in \{20, 40, 60, 80, 100\}$.
- $0 \leq A_i \leq 10^9$ for $i = 1, 2, \ldots, n$.
- $A_i > A_{i-1}$ for $1 < i \leq n$.

(**Warning**: For Python users, you are advised against using the standard dictionary data structure, which is rather slow.)

Test Cases

Your program will be evaluated on 5 visible test files and 5 hidden test files. Each test file is worth 0.6 points, and your program needs to answer all of the test instances correctly in a file to earn the points for that file.

Warning

There is a way to solve this problem without using only integer data types (e.g. `int`, `long long` in C++). To achieve this, you may want to scale all A_i 60 times. (Please be careful about overflow when scaling the number.) If you use floating-point numbers, please ensure that your program can handle precision errors properly.

Subtask

Out of the 10 test files, 8 of them satisfy the additional constraint:

- $p_i = 100$ for $i = 1, 2, \ldots, n$.

That is, all towers will have the same coverage radius d regardless of which cities they are built. With this additional constraint, it can be proven that the minimum coverage distance d must be an integer.

Sample Input 1:

```
5
5 1
100 100 100 100 100
1 3 5 7 9
5 2
100 100 100 100 100
1 3 5 7 9
5 3
100 100 100 100 100
1 3 5 7 9
5 4
100 100 100 100 100
1 3 5 7 9
5 5
100 100 100 100 100
1 3 5 7 9
```

Sample Output 1:

```
240
120
120
120
0
```

Sample Input 2:

```
2
6 2
60 20 20 20 20 80
10 20 30 110 120 130
2 1
20 20
0 1000000000
```

Sample Output 2:

```
2000
300000000000
```

Sample Explanation

Sample Input 1 contains $C = 4$ test instances. All test instances have the same $n = 5$, the same percentages $(p_1, p_2, p_3, p_4, p_5) = (100, 100, 100, 100, 100)$, and the same city locations: $(A_1, A_2, A_3, A_4, A_5) = (1, 3, 5, 7, 9)$. The only difference is the value of k. The outputs are explained as follows:

- For $k = 1$, the smallest coverage radius is $d = 4$, because we can build the tower at $A_3 = 5$. Note that the output is $60d = 240$.
- For $k = 2$, the smallest coverage radius is $d = 2$, because we can build two towers at $A_2 = 3$ and $A_4 = 7$.
- For $k = 3$ or 4, the smallest coverage radii are also $d = 2$.
- For $k = 5$, the smallest coverage radius is $d = 0$ because we can build one tower at each city.

For Sample Input 2, the solution is $d = 100/3$ for the first instance. To achieve this, we should build the towers at $A_1 = 10$ and $A_6 = 130$. The coverage ranges of the two towers are $d \cdot p_1/100 = 20$ and $d \cdot p_6/100 = 80/3$, respectively.

Submission Guideline

Write your program in either C, C++, or Python **in a single file**. Submit the file on Gradescope. The time limit on Gradescope is 4 seconds for C/C++ and 10 seconds for Python. You can make at most 10 submission attempts. **You may refer to or directly use the sample code files sample.cpp or sample.py that take input and write output.**