Problem 3386: Button with Longest Push Time

Problem Information

Difficulty: Easy
Acceptance Rate: 40.86%
Paid Only: No
Tags: Array

Problem Description

You are given a 2D array `events` which represents a sequence of events where a child pushes a series of buttons on a keyboard.

Each `events[i] = [indexi, timei]` indicates that the button at index `indexi` was pressed at time `timei`.

* The array is **sorted** in increasing order of `time`. * The time taken to press a button is the difference in time between consecutive button presses. The time for the first button is simply the time at which it was pressed.

Return the `index` of the button that took the **longest** time to push. If multiple buttons have the same longest time, return the button with the **smallest** `index`.

Example 1:

Input: events = [[1,2],[2,5],[3,9],[1,15]]

Output: 1

Explanation:

* Button with index 1 is pressed at time 2. * Button with index 2 is pressed at time 5, so it took `5 - 2 = 3` units of time. * Button with index 3 is pressed at time 9, so it took `9 - 5 = 4` units of time. * Button with index 1 is pressed again at time 15, so it took `15 - 9 = 6` units of time.

Example 2:

Input: events = [[10,5],[1,7]]

Output: 10

Explanation:

* Button with index 10 is pressed at time 5. * Button with index 1 is pressed at time 7, so it took `7 - 5 = 2` units of time.

Constraints:

* `1 <= events.length <= 1000` * `events[i] == [indexi, timei]` * `1 <= indexi, timei <= 105` * The input is generated such that `events` is sorted in increasing order of `timei`.

Code Snippets

C++:

```
class Solution {
public:
int buttonWithLongestTime(vector<vector<int>>& events) {

}
};
```

Java:

```
class Solution {
public int buttonWithLongestTime(int[][] events) {

}
}
```

Python3:

```
class Solution:
def buttonWithLongestTime(self, events: List[List[int]]) -> int:
```