

Problem 1629: Slowest Key

Problem Information

Difficulty: Easy

Acceptance Rate: 59.44%

Paid Only: No

Tags: Array, String

Problem Description

A newly designed keypad was tested, where a tester pressed a sequence of `n` keys, one at a time.

You are given a string `keyPressed` of length `n`, where `keyPressed[i]` was the `ith` key pressed in the testing sequence, and a sorted list `releaseTimes`, where `releaseTimes[i]` was the time the `ith` key was released. Both arrays are **0-indexed**. The `0th` key was pressed at the time `0`, and every subsequent key was pressed at the **exact** time the previous key was released.

The tester wants to know the key of the keypress that had the **longest duration**. The `ith` keypress had a **duration** of `releaseTimes[i] - releaseTimes[i - 1]`, and the `0th` keypress had a duration of `releaseTimes[0]` .

Note that the same key could have been pressed multiple times during the test, and these multiple presses of the same key **may not** have had the same **duration**.

Return the key of the keypress that had the**longest duration**. If there are multiple such keypresses, return the lexicographically largest key of the keypresses.

Example 1:

Input: releaseTimes = [9,29,49,50], keyPressed = "cbcd" **Output:** "c" **Explanation:**
The keypresses were as follows: Keypress for 'c' had a duration of 9 (pressed at time 0 and released at time 9). Keypress for 'b' had a duration of 29 - 9 = 20 (pressed at time 9 right after the release of the previous character and released at time 29). Keypress for 'c' had a duration of 49 - 29 = 20 (pressed at time 29 right after the release of the previous character and

released at time 49). Keypress for 'd' had a duration of $50 - 49 = 1$ (pressed at time 49 right after the release of the previous character and released at time 50). The longest of these was the keypress for 'b' and the second keypress for 'c', both with duration 20. 'c' is lexicographically larger than 'b', so the answer is 'c'.

Example 2:

Input: releaseTimes = [12,23,36,46,62], keysPressed = "spuda" **Output:** "a"

Explanation: The keypresses were as follows: Keypress for 's' had a duration of 12. Keypress for 'p' had a duration of $23 - 12 = 11$. Keypress for 'u' had a duration of $36 - 23 = 13$. Keypress for 'd' had a duration of $46 - 36 = 10$. Keypress for 'a' had a duration of $62 - 46 = 16$. The longest of these was the keypress for 'a' with duration 16.

Constraints:

* `releaseTimes.length == n` * `keysPressed.length == n` * `2 <= n <= 1000` * `1 <= releaseTimes[i] <= 109` * `releaseTimes[i] < releaseTimes[i+1]` * `keysPressed` contains only lowercase English letters.

Code Snippets

C++:

```
class Solution {
public:
    char slowestKey(vector<int>& releaseTimes, string keysPressed) {
        }
};
```

Java:

```
class Solution {
public char slowestKey(int[] releaseTimes, String keysPressed) {
        }
}
```

Python3:

```
class Solution:  
    def slowestKey(self, releaseTimes: List[int], keysPressed: str) -> str:
```