SLOWEST KEY PRESS

https://leetcode.com/problems/slowest-key/discuss/909278/Java-0ms-100-AC-Solution-O(n)-Cracked

class Solution {

public char slowestKey(int[] releaseTimes, String keysPressed) {

int slowestKeyCount = releaseTimes[0];

char slowestKey = keysPressed.charAt(0);

for(int i=1; i<releaseTimes.length; i++) {

int diff = releaseTimes[i] - releaseTimes[i-1];

if(diff > slowestKeyCount) {

slowestKeyCount = diff;

slowestKey = keysPressed.charAt(i);

}

else if(diff == slowestKeyCount) {

slowestKey = keysPressed.charAt(i) > slowestKey ? keysPressed.charAt(i) : slowestKey;

}

}

return slowestKey;

}

}

Alternate :

class Solution {

public char slowestKey(int[] releaseTimes, String keysPressed) {

int longestDuration = releaseTimes[0];

char slowestKey = keysPressed.charAt(0);

for (int i = 1; i < releaseTimes.length; i++) {

int duration = releaseTimes[i] - releaseTimes[i - 1];

if (duration > longestDuration || (duration == longestDuration && keysPressed.charAt(i) > slowestKey)) {

slowestKey = keysPressed.charAt(i);

longestDuration = duration;

}

}

return slowestKey;

}

}

Alternate 3 :

class Solution {

public char slowestKey(int[] releaseTimes, String keysPressed) {

int[] arr = new int[releaseTimes.length];

arr[0] = releaseTimes[0];

int max =arr[0];

int res =keysPressed.charAt(0)-'a';

for(int i =1;i<releaseTimes.length;i++){

arr[i] = releaseTimes[i] -releaseTimes[i-1];

if(arr[i] == max){

res = Math.max(res,keysPressed.charAt(i)-'a');

}

else if(arr[i] > max){

max = arr[i];

res = keysPressed.charAt(i)-'a';

}

}

return (char)(res+'a');

}

}