



# [CS 11] Prac 9j – Savings

## Problem Statement

You diligently maintain a "piggy bank" of coins. From time to time, whenever you have spare coins, you sometimes put them in the piggy bank.

However, instead of an actual piggy bank, you simply use an unsealed, openable plastic jar. This makes it impossible to actually save money, because the system is vulnerable to theft! And in fact, the thief is **you**—you regularly get tempted to "withdraw" some coins from the piggy bank, because nothing is really stopping you from doing so.

To make yourself feel better, you have decided to collect the periods of time when you *didn't* take away anything from the piggy bank. Because the jar is placed on top of a weighing scale which logs the weight of the bank over time, you have precise hourly data on the number of coins in the bank.

Given the hourly data, slice it up into maximal contiguous *sorted* subsequences. For example, if the input sequence is

- 3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5, 8, 9, 7, 9, 3, 3

then the subsequences we're looking for are:

- 3
- 1, 4
- 1, 5, 9
- 2, 6
- 5
- 3, 5, 8, 9
- 7, 9
- 3, 3

## Task Details

Your task is to implement a function called `sorted_streaks`. This function has a single parameters, a `tuple` or `list` of `int`s.

The function must return a `list` of `tuple`s of `int`s representing the maximal contiguous sorted subsequences.

## Restrictions

(See 9a for more restrictions)

For this problem in particular:

- Recursion is allowed.
- The source code limit is 2000.

## Example Calls

### Example 1 Function Call

```
sorted_streaks((3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5, 8, 9, 7, 9,  
3))
```

Copy

### Example 1 Return Value

```
[  
 (3,),  
 (1, 4),  
 (1, 5, 9),  
 (2, 6),  
 (5,),  
 (3, 5, 8, 9),  
 (7, 9),  
 (3, 3),  
)
```

Copy

## Constraints

- The function `sorted_streaks` will be called at most 60,000 times.
- The length of each input sequence will be at most 200,000.
- The total length of all input sequences will be at most 200,000.
- Each integer will be between 0 and  $10^{10}$ .

## Scoring

- You get 45 ❤ points if you solve all test cases where:
  - The length of each sequence is at most has length at most 50.
  - The total length of all input sequences is at most 500.

- You get 40 ❤ points if you solve all test cases where:
  - The length of each sequence is at most has length at most 4,000.
  - The total length of all input sequences is at most 8,000.

- You get 60 ❤ points if you solve all test cases.

## ?

## Clarifications

Report an issue

No clarifications have been made at this time.