



# [CS 11] Prac 10i – Weakest Link

## Problem Statement

Steve has just learned how to make chainmail armor, and he now wants to make one.

He needs several *chains* to make chainmail armor, so he went to the local *chain store* to buy some. The chain store sells chains in a weird way. First, they display  $n$  *links* in a row in front of their store. Then, to buy a chain, you select a *contiguous* subsequence of the links. The store will make copies of those links for you and turn them into a chain. You then pay with cash—credit cards are not accepted.

More formally, if we number the links 1 to  $n$  from left to right, then you buy a chain by selecting two links  $i$  and  $j$  with  $1 \leq i \leq j \leq n$ , and the store will make a chain out of copies of links  $i$  to  $j$ , inclusive.

Thus, you can buy  $n(n + 1)/2$  distinct chains from the store. Steve went ahead and bought exactly one of each distinct chain, so he now has  $n(n + 1)/2$  chains.

However, he soon realized that this was a bad idea—most of the chains are weak, because a chain is only as strong as its *weakest link*! More formally, if link  $i$  has a strength of  $s_i$ , then the strength of the chain formed from links  $i$  to  $j$  is the minimum among  $s_i, s_{i+1}, \dots, s_j$ .

What is the strength of the  $k$ th strongest chain in Steve's collection?

## Task Details

Your task is to implement a function called `kth_strongest_chain`. This function has two positional arguments:

- the first is the `int`  $k$ ;
- the second is a `tuple` / `list` of  $n$  `int`'s representing the strength values (" $s_i$ " above).

The function must return an `int` denoting the strength of the  $k$ th strongest chain.

## Restrictions

(See 10a for more restrictions)

For this problem in particular:

- The following symbols are allowed: `map`, `filter`.
- The following import is allowed: `cache` and `lru_cache` from `functools`.
- The source code limit is 1000.

## Example Calls

### Example 1 Function Call

```
kth_strongest_chain(3, (4, 6, 6, 9, 2, 0, 1, 6))
```

Copy

### Example 1 Return Value

```
6
```

Copy

### Example 2 Function Call

```
kth_strongest_chain(1, (4, 6, 6, 9, 2, 0, 1, 6))
```

Copy

### Example 2 Return Value

```
9
```

Copy

### Example 3 Function Call

```
kth_strongest_chain(11, [4, 6, 6, 9, 2, 0, 1, 6])
```

Copy

### Example 3 Return Value

```
4
```

Copy

## Constraints

- The function `kth_strongest_chain` will be called at most 60,000 times.
- The sum of  $ns$  across all calls will be  $\leq 250,000$ .
- $1 \leq n \leq 250,000$
- $1 \leq k \leq n(n + 1)/2$
- Each strength value is an integer between 0 and  $10^{10}$ .

## Scoring

- You get 55 ❤ points if you solve all test cases where:
  - $n \leq 50$
  - The sum of the  $ns$  across all calls is 500.
- You get 50 ❤ points if you solve all test cases where:
  - $n \leq 4,000$
  - The sum of the  $ns$  across all calls is 8,000.
- You get 65 ❤ points if you solve all test cases.

## ?

## Clarifications

Report an issue

No clarifications have been made at this time.