[CS 11 25.1] Lab 3i – 寿司 2

Cheatsheet is available here: https://oj.dcs.upd.edu.ph/cs11cheatsheet/

Problem Statement

You are back at the same sushi restaurant! Once again, there are n plates of sushi that are going to pass by your table, but this time, the chef has told you in advance how many pieces of sushi each plate is going to have!

Each plate of sushi contains either 1, 2, or 4 pieces of sushi.

You do not want to disturb the other customers too much, so you decide to get **at most k consecutive** plates of sushi.

If you choose which plates of sushi you get optimally, at most how many pieces of sushi can you get?

Task Details

Your task is to implement a function named `max_pieces`, which should have the following *signature*:

```
def max_pieces(conveyor_belt, k):
```

Copy

The above says that it has two arguments `conveyor_belt` and k.

- `conveyor_belt` is a `list` of integers (`int`s) denoting the how many pieces of sushi are on each plate of the conveyor belt. Note that the first element of `conveyor_belt` corresponds to the first plate of sushi that will pass by your table.
- k is an integer.

The function must return an integer denoting the maximum number of pieces of sushi that you can get.

Restrictions

- The following symbols can now be used: `list`, `set`, `dict`, `enumerate`, `append`, `pop`, `extend`, `remove`, `sort`, `sorted`, `insert`, `clear`, `reverse`, `reversed`.
- Loops are allowed.
- Recursion is *disallowed*.
- Comprehensions are *disallowed*.
- Your source code must have at most 600 bytes.

Examples

Example 1 Function Call

```
max_pieces((1, 4, 2, 1, 4, 2, 2, 1, 4, 2), 5)
```

Copy

Example 1 Return Value

```
13
```

Copy

Constraints

- The function `max_pieces` will be called at most $70{,}000$ times.
- $0 \le k \le n \le 350{,}000$
- The sum of n across all calls to `max_pieces` will be $\le 350{,}000$.
- Each plate of sushi contains either 1, 2, or 4 pieces of sushi.

Scoring

Note: New tests may be added and all submissions may be rejudged at a later time. (All future tests will satisfy the constraints.)

- You get 45 ❤️ points if you solve all test cases where:
  - $n \le 50$
  - The sum of n across all calls to `max_pieces` will be ≤ 600.
- You get 100 🔴 points if you solve all test cases where:
  - $n \le 5{,}000$
  - The sum of n across all calls to `max_pieces` will be $\le 10{,}000$.
- You get 25 💔 points if you solve all test cases.

Clarifications

Report an issue

No clarifications have been made at this time.

Submit solution [CS 11 25.1] Lab Exercise 3

My submissions

✓ **Points:** 170 (partial)

Time limit: 7.0s

Memory limit: 2G

› **Problem type**

⌄ **Allowed languages**

py3