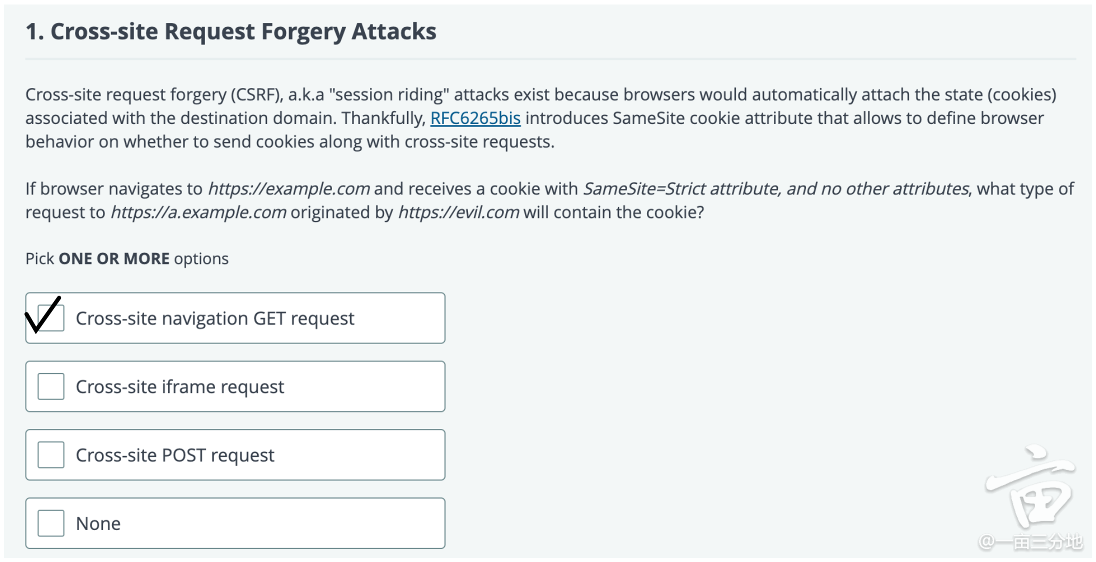


1. Cross-site Request Forgery Attacks

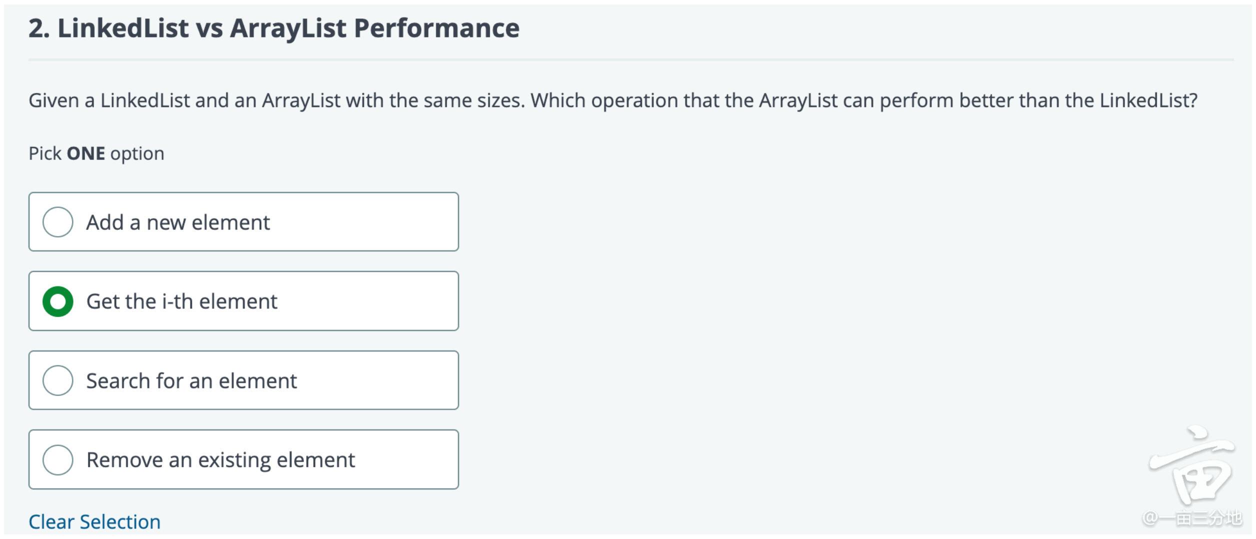
Cross-site request forgery (CSRF), a.k.a "session riding" attacks exist because browsers would automatically attach the state (cookies) associated with the destination domain. Thankfully, RFC6265bis introduces SameSite cookie attribute that allows to define browser behavior on whether to send cookies along with cross-site requests.

If browser navigates to *https://example.com* and receives a cookie with *SameSite=Strict attribute, and no other attributes*, what type of request to *https://a.example.com* originated by *https://evil.com* will contain the cookie?

Pick **ONE OR MORE** options

- [✓ Cross-site navigation GET request]
- ☐ Cross-site iframe request
- ☐ Cross-site POST request
- ☐ None

@一亩三分地



2. LinkedList vs ArrayList Performance

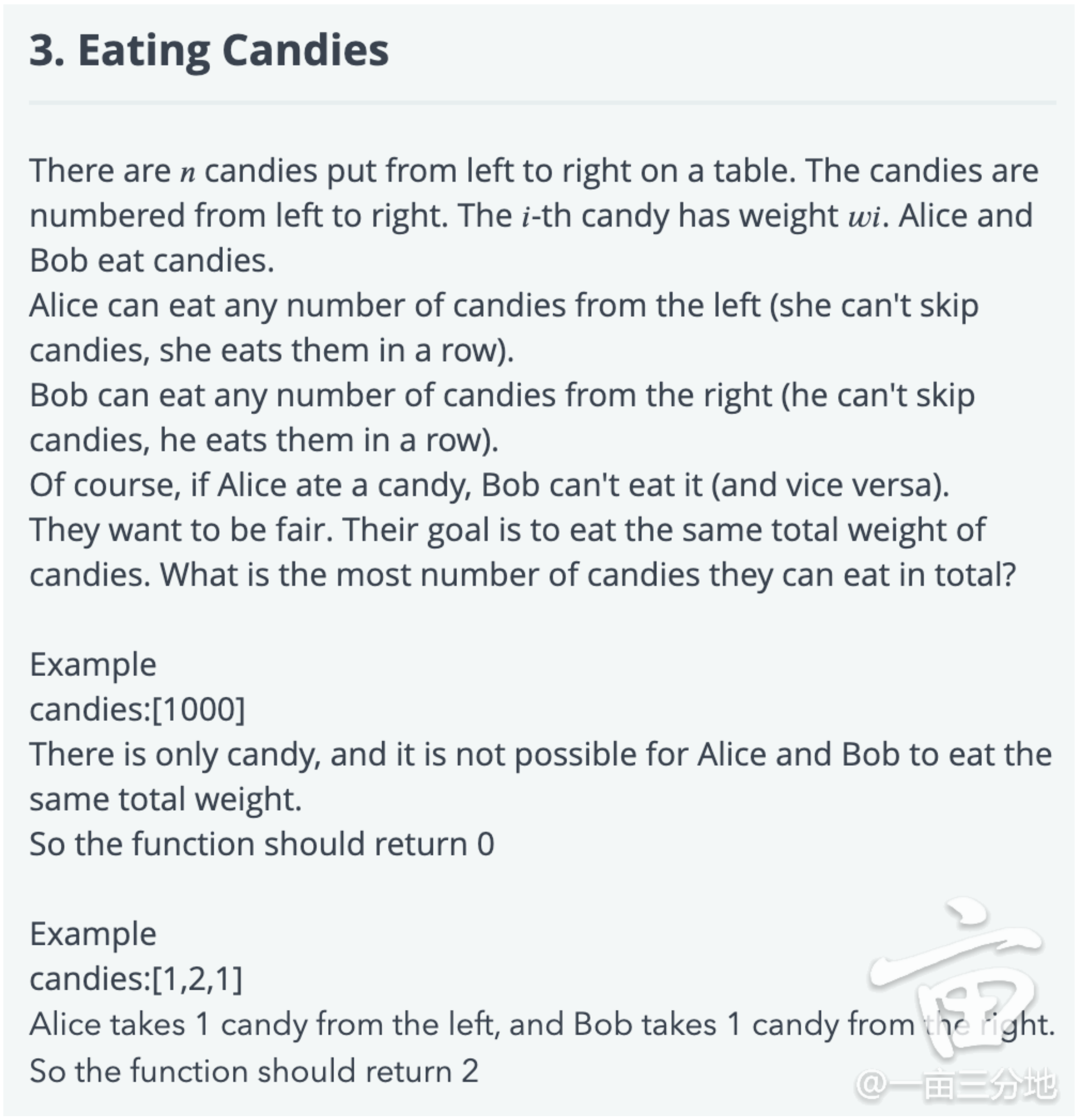
Given a LinkedList and an ArrayList with the same sizes. Which operation that the ArrayList can perform better than the LinkedList?

Pick **ONE** option

- ○ Add a new element
- [✓ Get the i-th element]
- ○ Search for an element
- ○ Remove an existing element

Clear Selection

@一亩三分地



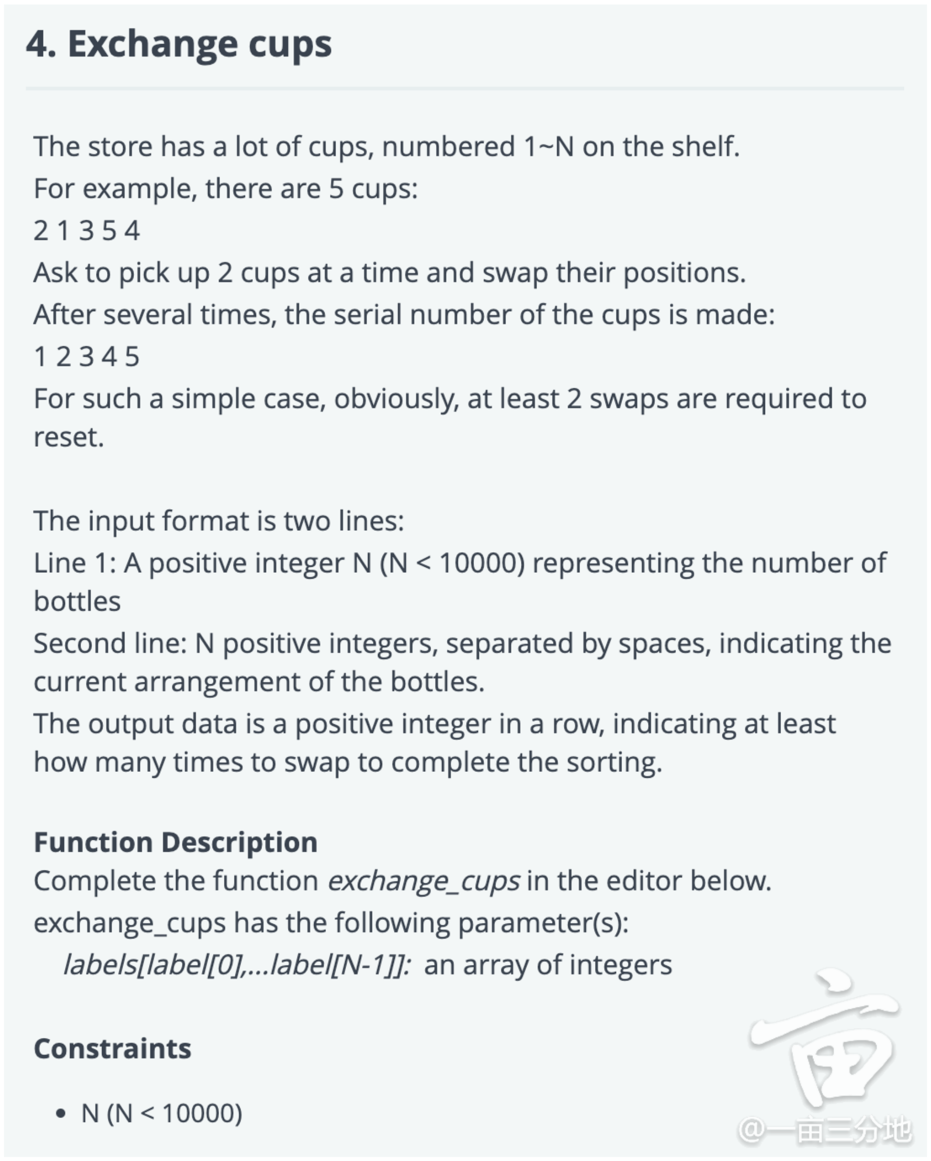
3. Eating Candies

There are n candies put from left to right on a table. The candies are numbered from left to right. The i-th candy has weight wi. Alice and Bob eat candies.
Alice can eat any number of candies from the left (she can't skip candies, she eats them in a row).
Bob can eat any number of candies from the right (he can't skip candies, he eats them in a row).
Of course, if Alice ate a candy, Bob can't eat it (and vice versa).
They want to be fair. Their goal is to eat the same total weight of candies. What is the most number of candies they can eat in total?

Example
candies:[1000]
There is only candy, and it is not possible for Alice and Bob to eat the same total weight.
So the function should return 0

Example
candies:[1,2,1]
Alice takes 1 candy from the left, and Bob takes 1 candy from the right.
So the function should return 2

@一亩三分地



4. Exchange cups

The store has a lot of cups, numbered 1~N on the shelf.
For example, there are 5 cups:
2 1 3 5 4
Ask to pick up 2 cups at a time and swap their positions.
After several times, the serial number of the cups is made:
1 2 3 4 5
For such a simple case, obviously, at least 2 swaps are required to reset.

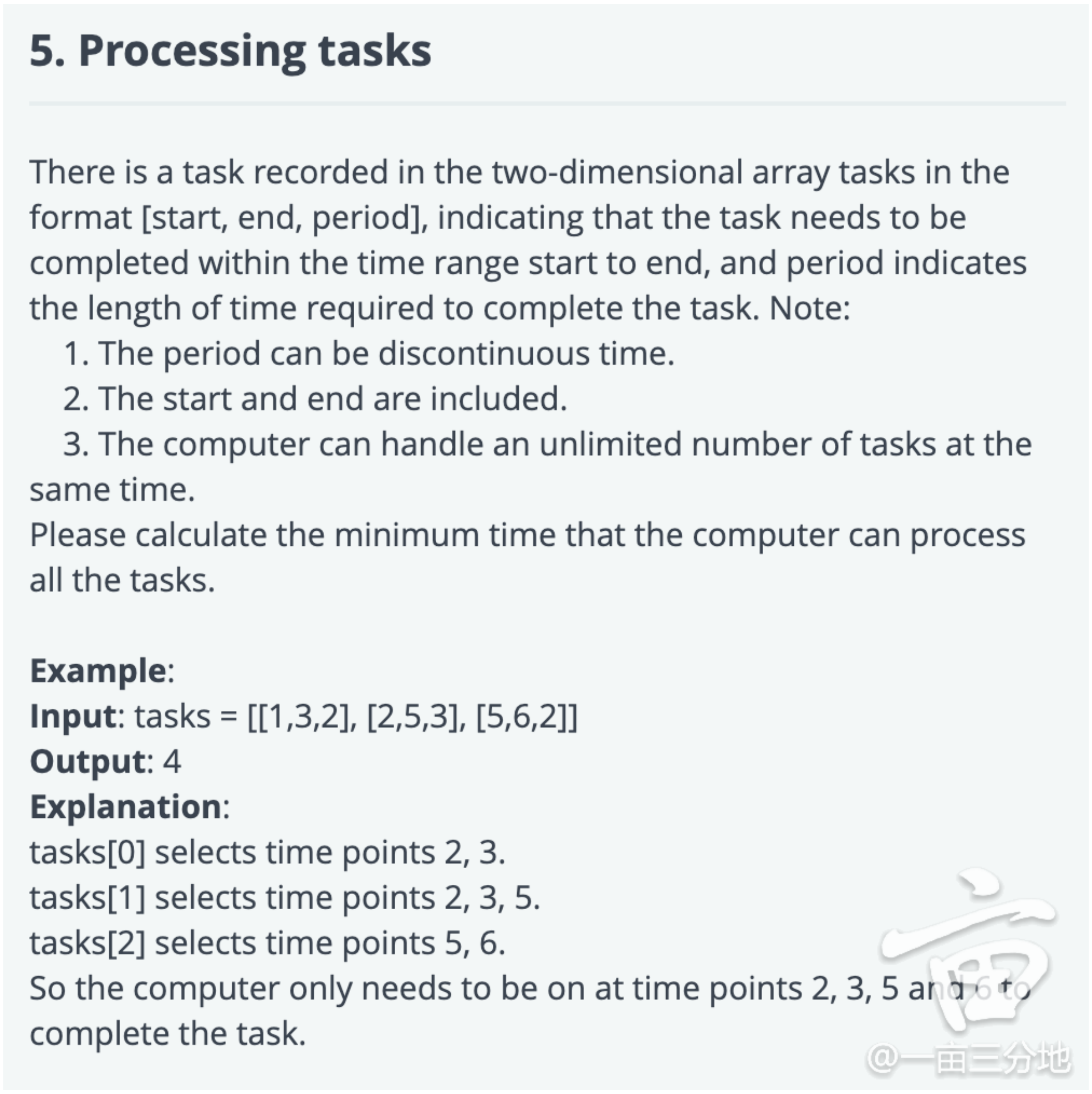
The input format is two lines:
Line 1: A positive integer N (N < 10000) representing the number of bottles
Second line: N positive integers, separated by spaces, indicating the current arrangement of the bottles.
The output data is a positive integer in a row, indicating at least how many times to swap to complete the sorting.

Function Description
Complete the function *exchange_cups* in the editor below.
exchange_cups has the following parameter(s):
  labels[label[0],...label[N-1]]: an array of integers

Constraints

- N (N < 10000)

亩 @一亩三分地



5. Processing tasks

There is a task recorded in the two-dimensional array tasks in the format [start, end, period], indicating that the task needs to be completed within the time range start to end, and period indicates the length of time required to complete the task. Note:

1. The period can be discontinuous time.
2. The start and end are included.
3. The computer can handle an unlimited number of tasks at the same time.

Please calculate the minimum time that the computer can process all the tasks.

Example:
Input: tasks = [[1,3,2], [2,5,3], [5,6,2]]
Output: 4
Explanation:
tasks[0] selects time points 2, 3.
tasks[1] selects time points 2, 3, 5.
tasks[2] selects time points 5, 6.
So the computer only needs to be on at time points 2, 3, 5 and 6 to complete the task.

@一亩三分地

https://www.luogu.com.cn/problem/P1250