

1700. Number of Students Unable to Eat Lunch

[My Submissions \(/contest/biweekly-contest-42/problems/number-of-students-unable-to-eat-lunch/submissions/\)](#)

[Back to Contest \(/contest/biweekly-contest-42/\)](#)

The school cafeteria offers circular and square sandwiches at lunch break, referred to by numbers `0` and `1` respectively. All students stand in a queue. Each student either prefers square or circular sandwiches.

The number of sandwiches in the cafeteria is equal to the number of students. The sandwiches are placed in a **stack**. At each step:

- If the student at the front of the queue **prefers** the sandwich on the top of the stack, they will **take it** and leave the queue.
- Otherwise, they will **leave it** and go to the queue's end.

This continues until none of the queue students want to take the top sandwich and are thus unable to eat.

You are given two integer arrays `students` and `sandwiches` where `sandwiches[i]` is the type of the i^{th} sandwich in the stack ($i = 0$ is the top of the stack) and `students[j]` is the preference of the j^{th} student in the initial queue ($j = 0$ is the front of the queue). Return *the number of students that are unable to eat*.

User Accepted:	2823
User Tried:	3119
Total Accepted:	2922
Total Submissions:	4448
Difficulty:	Easy

Example 1:

Input: `students = [1,1,0,0]`, `sandwiches = [0,1,0,1]`

Output: `0`

Explanation:

- Front student leaves the top sandwich and returns to the end of the line making `students = [1,0,0,1]`.
- Front student leaves the top sandwich and returns to the end of the line making `students = [0,0,1,1]`.
- Front student takes the top sandwich and leaves the line making `students = [0,1,1]` and `sandwiches = [1,0,1]`.
- Front student leaves the top sandwich and returns to the end of the line making `students = [1,1,0]`.
- Front student takes the top sandwich and leaves the line making `students = [1,0]` and `sandwiches = [0,1]`.
- Front student leaves the top sandwich and returns to the end of the line making `students = [0,1]`.
- Front student takes the top sandwich and leaves the line making `students = [1]` and `sandwiches = [1]`.
- Front student takes the top sandwich and leaves the line making `students = []` and `sandwiches = []`.

Hence all students are able to eat.

Example 2:

Input: `students = [1,1,1,0,0,1]`, `sandwiches = [1,0,0,0,1,1]`

Output: `3`

Constraints:

- `1 <= students.length, sandwiches.length <= 100`
- `students.length == sandwiches.length`
- `sandwiches[i]` is `0` or `1`.
- `students[i]` is `0` or `1`.

[Discuss \(https://leetcode.com/problems/number-of-students-unable-to-eat-lunch/discuss\)](#)

Java

```
1 class Solution {
2     public int countStudents(int[] students, int[] sandwiches) {
3         int[] countZeroAndOne = new int[2];
4         for (int i: students) countZeroAndOne[i]++;
5         for (int i=0;i<sandwiches.length;i++)
6             if (--countZeroAndOne[sandwiches[i]] < 0) return sandwiches.length - i;
7         return 0;
8     }
9 }
10
11 }
```