



Alice, Bob and Chocolate

Link submit: <http://codeforces.com/problemset/problem/6/C>

Solution:

C++	https://ideone.com/sZsMvR
Java	https://ideone.com/I5XVSt
Python	https://ideone.com/hGHTQ0

Tóm tắt đề:

Có n thanh chocolate trên bàn. Khi trò chơi bắt đầu, Alice sẽ ăn các thanh chocolate lần lượt từ trái sang phải, trong khi Bob sẽ ăn theo chiều ngược lại. Nếu một thanh chocolate mà cả hai cùng bắt đầu ăn vào một thời điểm thì Bob sẽ nhường cho Alice.

Tìm số thanh chocolate mà Alice và Bob ăn được.

Input:

Dòng đầu chứa một số nguyên dương N là số thanh chocolate.

Dòng thứ hai là một dãy gồm N số t_1, t_2, \dots, t_n ($1 \leq t_i \leq 1000$) với t_i là thời gian cần để ăn hết thanh chocolate thứ i .

Output:

In ra lần lượt là số thanh chocolate Alice và Bob đã ăn.

Ví dụ:

5	2 3
2 9 8 2 7	

Giải thích ví dụ:

Alice sẽ ăn lần lượt các thanh số 1, 2, 3 (tương ứng với thời gian là 2, 9, 8) và Bob sẽ ăn lần lượt thanh số 5, 4 (có thời gian lần lượt là 7, 2).

Hướng dẫn giải:

Bỏ tất cả các phần tử vào vector v . Gọt t_1, t_2 lần lượt là thời điểm bắt đầu ăn của Alice và Bob.

Sử dụng kỹ thuật two pointers để giải quyết bài toán: biến l chạy từ đầu mảng, r chạy từ cuối mảng.

Trong khi 2 pointers chưa gặp nhau ($l \leq r$), ta xét xem:

- Nếu thời điểm ăn của Alice (tức $t1 + v[l]$) nhỏ hơn **hoặc bằng** thời điểm ăn của Bob (tức $t2 + v[r]$) thì sẽ cập nhật lại thời điểm $t1$ và tăng l lên 1.
- Ngược lại, cập nhật lại thời điểm $t2$ và giảm r đi 1.

Số thanh chocolate Alice ăn được chính là vị trí l và số thanh Bob ăn được là $(n - l)$.

Độ phức tạp: $O(N)$ với N là số lượng thanh chocolate.

Big-O Coding