



## Approximating a Constant Range

Link submit: <http://codeforces.com/problemset/problem/602/B>

Solution:

C++	<a href="https://ideone.com/uqrSFu">https://ideone.com/uqrSFu</a>
Java	<a href="https://ideone.com/QQ4v8o">https://ideone.com/QQ4v8o</a>
Python	<a href="https://ideone.com/70NyhH">https://ideone.com/70NyhH</a>

Tóm tắt đề:

Khi Xellos thực hiện khóa học thực hành tại trường đại học, anh đã đo cường độ của một hiệu ứng đang dần tiến đến sự cân bằng. Cách tốt để xác định cường độ cân bằng là chọn một số lượng lớn các điểm dữ liệu liên tục nhất có thể và lấy trung bình của chúng. Tuy nhiên, với kích thước thông thường của dữ liệu nó không có gì khó khăn – nhưng tại sao chúng ta không đặt ra một bài toán lập trình tương tự khi đang nghiên cứu nó?

Bạn được cho một dãy liên tục gồm  $n$  điểm dữ liệu  $a_1, \dots, a_n$ . Nó không có bất kỳ bước nhảy lớn nào giữa các điểm dữ liệu liên tục – với mọi  $1 \leq i < n$ , nó đảm bảo rằng  $|a_{i+1} - a_i| \leq 1$ .

Một khoảng  $[l, r]$  của các điểm dữ liệu được cho là gần như cố định nếu như độ chênh lệch giữa giá trị lớn nhất và nhỏ nhất trong dãy đó không vượt quá 1. Gọi  $M$  là giá trị lớn nhất và  $m$  là giá trị nhỏ nhất của  $a_i$  với  $l \leq i \leq r$ , khoảng  $[l, r]$  gần như cố định nếu  $M - m \leq 1$ . Tìm chiều dài lớn nhất của khoảng gần như cố định.

Input:

Dòng đầu tiên là một số nguyên  $n$  ( $2 \leq n \leq 100\,000$ ) – số lượng các điểm dữ liệu.

Dòng thứ hai gồm  $n$  số nguyên  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 100\,000$ ).

Output:

Một số duy nhất – Chiều dài lớn nhất của một khoảng gần như cố định.

Ví dụ:

5 1 2 3 3 2	4
11 5 4 5 5 6 7 8 8 8 7 6	5

**Giải thích ví dụ:**

Trong ví dụ đầu tiên, khoảng gần như cố định dài nhất là  $[2, 5]$ , có chiều dài là 4 (số lượng các điểm dữ liệu).

Trong ví dụ thứ 2, có 3 dãy gần như không đổi có chiều dài là 4:  $[1, 4]$ ,  $[6, 9]$  và  $[7, 10]$ . Khoảng gần như cố định dài nhất là  $[6, 10]$  với chiều dài là 5.

**Hướng dẫn giải:**

Ta sẽ dùng một mảng  $cnt$  với  $cnt[i]$  là số lượng phần tử có giá trị  $i$  và một biến  $count\_diff$  đếm số giá trị phân biệt xuất hiện trong đoạn  $[l, r]$  đang xét.

Dùng kỹ thuật two pointer để xác định các đoạn  $[l, r]$ , ta sẽ duyệt theo  $r$ , với mỗi  $r$ , cập nhật số lần xuất hiện của  $a[r]$ , nếu  $a[r]$  là một giá trị mới vừa xuất hiện trong đoạn thì tăng  $count\_diff$  lên 1 đơn vị.

Nếu  $count\_diff$  lớn hơn 2 thì ta sẽ bắt đầu tăng  $l$  để loại bỏ đi các phần tử không hợp lệ (vì chỉ cho phép chênh lệch tối đa là 1, tức chỉ có 2 giá trị khác nhau xuất hiện trong đoạn).

Sau khi đã cập nhật lại giá trị  $l$  thì ta sẽ có đoạn  $[l, r]$  chính là đoạn dài nhất thỏa yêu cầu đề bài và kết thúc tại  $r$ . Ta cập nhật lại kết quả bài toán.

**Độ phức tạp:**  $O(N)$  với  $N$  là số phần tử của mảng.