

Approximating a Constant Range

Blue 11 - Lecture 02: Algorithmic Complexity

Tóm Tắt Đề Bài

Tóm Tắt Đề Bài

Dữ kiện:

- Cho một dãy **A** gồm **N** phần tử
- Các phần tử trong dãy **A** đều có tính chất $|A[i+1] - A[i]| \leq 1$
- Một đoạn **[L, R]** được gọi là gần như không đổi nếu như giá trị lớn nhất và giá trị nhỏ nhất trong đoạn có khoảng cách ≤ 1 (i.e $\max(A[L..R]) - \min(A[L..R]) \leq 1$)

Yêu cầu:

- Hãy tìm đoạn **[L, R]** gần như không đổi dài nhất có thể.

Giới hạn:

- $N \leq 100000$
- $A[i] \leq 100000$

Giải Thích Ví Dụ

Giải Thích Ví Dụ

Input	Output	Giải thích ví dụ
$N = 5$ $A = [1 \text{ } 2 \text{ } 3 \text{ } 3 \text{ } 2]$	4	Đoạn dài nhất gần như không đổi là đoạn $[2,5]$ có độ dài là 4 vì giá trị lớn nhất là 3, giá trị nhỏ nhất là 2 vậy nên thoả mãn điều kiện gần như không đổi

Giải Thích Ví Dụ

Input	Output	Giải thích ví dụ
$N = 11$ $A = [5\ 4\ 5\ 5\ 6\ 7\ 8\ 8\ 8\ 7\ 6]$	5	Các đoạn $[1, 4]$, $[6, 9]$, $[7, 10]$ đều là các đoạn gần như không đổi, tuy nhiên đối với đoạn $[6, 10]$ mới là đoạn gần như không đổi dài nhất và có độ dài là 5.

Hướng Dẫn Giải

Ý Tưởng

- Với mỗi phần tử thứ i :
 - Ta cần tìm phần tử thứ j **lớn nhất** sao cho $\max(A[i..j]) - \min(A[i..j]) \leq 1$
 - Cập nhật $\text{best} = \max(\text{best}, j - i + 1)$
- Do tính chất $|A[i+1] - A[i]| \leq 1$ của dãy A , ta nhận thấy rằng đoạn được xem là không thay đổi khi và chỉ khi đoạn đó có **không quá 2 phần tử phân biệt**.

5	6	7	7	8	8	7
---	---	---	---	---	---	---

Ý Tưởng

- Với mỗi phần tử thứ i :
 - Ta cần tìm phần tử thứ **j lớn nhất** sao cho **không quá 2 phần tử phân biệt**
 - Cập nhật **$best = \max(best, j - i + 1)$**
- Giả sử với vị trí i ta đã tìm **j xa nhất thoả mãn**, vậy với vị trí $i + 1$ thì **j xa nhất** là bao nhiêu ?
 - Gọi **j xa nhất** của vị trí $i + 1$ là **j'**
 - Ta có nhận xét là **$j \leq j'$**
 - Sử dụng kỹ thuật **hai con trỏ** để lưu **vị trí cuối cùng** của j và tịnh tiến lên một vài vị trí để tính được **j'**

Ý Tưởng

- Vấn đề là làm sao chúng ta có thể check được đoạn $[i, j]$ vừa tìm có thoả mãn tính chất chỉ chứa không quá 2 phần tử khác nhau ?
- Ta sử dụng kỹ thuật mảng *fre* đếm tần suất của các số trong đoạn $[i, j]$
 - Trường hợp ta nói rộng đoạn $[i, j] \rightarrow [i, j + 1]$:
 - Xét $A[j + 1]$ có $fre[A[j + 1]] = 0$, tức là số $A[j+1]$ trong đoạn $[i, j]$ chưa xuất hiện do số lần xuất hiện của nó = 0
 - Khi nói ra $j + 1$ nó sẽ làm số lượng số khác nhau trong mảng thêm 1.
 - Trường hợp ta thu hẹp đoạn $[i, j] \rightarrow [i + 1, j]$:
 - Xét $A[i]$ có $fre[A[i]] = 1$, tức là số $A[i]$ trong đoạn chỉ xuất hiện một lần
 - Khi thu hẹp $i+1$, nó sẽ làm số lượng số khác nhau trong mảng giảm đi 1.

Ý Tưởng

5	6	7	7	8	8	7
---	---	---	---	---	---	---

Count = 0

	1	2	3	4	5	6	7	8
<i>fre</i>	0	0	0	0	0	0	0	0

Ý Tưởng

5	6	7	7	8	8	7
---	---	---	---	---	---	---




Count = 1

	1	2	3	4	5	6	7	8
<i>fre</i>	0	0	0	0	1	0	0	0

Ý Tưởng

5	6	7	7	8	8	7
---	---	---	---	---	---	---




Count = 2

	1	2	3	4	5	6	7	8
<i>fre</i>	0	0	0	0	1	1	0	0

Ý Tưởng

5	6	7	7	8	8	7
---	---	---	---	---	---	---




Count = 1

	1	2	3	4	5	6	7	8
<i>fre</i>	0	0	0	0	0	1	0	0

Ý Tưởng

5	6	7	7	8	8	7
---	---	---	---	---	---	---




Count = 2

	1	2	3	4	5	6	7	8
<i>fre</i>	0	0	0	0	0	1	1	0

Ý Tưởng

5	6	7	7	8	8	7
---	---	---	---	---	---	---




Count = 2

	1	2	3	4	5	6	7	8
<i>fre</i>	0	0	0	0	0	1	2	0

Ý Tưởng

5	6	7	7	8	8	7
---	---	---	---	---	---	---




Count = 1

	1	2	3	4	5	6	7	8
<i>fre</i>	0	0	0	0	0	0	2	0

Ý Tưởng

5	6	7	7	8	8	7
---	---	---	---	---	---	---




Count = 2

	1	2	3	4	5	6	7	8
<i>fre</i>	0	0	0	0	0	0	2	1

Ý Tưởng

5	6	7	7	8	8	7
---	---	---	---	---	---	---




Count = 2

	1	2	3	4	5	6	7	8
<i>fre</i>	0	0	0	0	0	0	2	2

Ý Tưởng

5	6	7	7	8	8	7
---	---	---	---	---	---	---




Count = 2

	1	2	3	4	5	6	7	8
<i>fre</i>	0	0	0	0	0	0	3	2

Ý Tưởng

5	6	7	7	8	8	7
---	---	---	---	---	---	---



Count = 2

	1	2	3	4	5	6	7	8
<i>fre</i>	0	0	0	0	0	0	2	2

Thuật Toán

- Bước 1: Đọc N và mảng A .
- Bước 2: Khởi tạo
 - **right** = 0, **best** = 0, **count** = 0
 - Mảng *fre* có 100000 phần tử 0
- Bước 3: Duyệt **left** = 0 tới $N - 1$
 - Ta cần tìm vị trí **right** xa nhất sao cho **count** ≤ 2 :
 - Nếu **count** == 2 và $\text{fre}[A[\text{right}]] == 0$, ta thoát khỏi vòng lặp
 - Nếu $\text{fre}[A[\text{right}]] == 0$, ta cập nhật **count** += 1
 - $\text{fre}[A[\text{right}]] += 1$
 - **right** += 1
 - Cập nhật **best** = **max(right - left, best)**
 - Chuẩn bị sang vị trí kế tiếp, ta cập nhật:
 - Nếu $\text{fre}[A[\text{left}]] == 1$, ta cập nhật **count** -= 1
 - $\text{fre}[A[\text{left}]] -= 1$
- Bước 4: Đưa ra kết quả **best**

Độ phức tạp: **$O(N)$**

Mã Giả

Pseudo Code

```
read(N, A)
right = 0, best = 0, count = 0
fre = [0] * (10^5+1)
for left := 0 to N - 1 do:
    while right <= N and count <= 2:
        if (count == 2) and (fre[A[right]] == 0):
            break
        if (fre[A[right]] == 0):
            count += 1
        fre[A[right]] += 1
        right += 1
    best = max(best, right - left)
    if (fre[A[left]] == 1):
        count -= 1
    fre[A[left]] -= 1
write(best)
```


Thank you