DPC7

Nguồn: Educational Codeforces Round 63 - Problem D

https://codeforces.com/contest/1155/problem/D

Với một dãy A gồm n phần tử, ta có thể tính giá trị $f(A)$ với độ phức tạp $O(n)$ bằng thuật toán Kadane (các bạn có thể tham khảo thêm về thuật toán Kadane trong bài viết sau: https://cp-algorithms.com/others/maximum_average_segment.html)

Ta có thể giải bài toán này bằng phương pháp quy hoạch động, kết hợp với ý tưởng thuật toán Kadane:

Gọi $dp[i][state_{mul}]$ là tổng của đoạn con có tổng lớn nhất kết thúc tại vị trí i, $state_{mul}$ là trạng thái tại vị trí i của đoạn con được nhân với x ($state_{mul} = 0$ nếu đoạn con đó chưa được mở, $state_{mul} = 1$ nếu đoạn con đó đang được mở và sẽ nhận thêm phần tử a_i, $state_{mul} = 2$ nếu đoạn con đó đã được đóng và không nhận thêm phần tử).

Tại mỗi bước, với trạng thái $(i, state_{mul})$, ta có hai bước chuyển trạng thái:

- Thay đổi trạng thái đoạn được nhân với x, từ chưa mở thành đang được mở ($state_mul$ từ 0 thành 1) hoặc từ đang được mở thành đã đóng ($state_mul$ từ 1 thành 2) mà không chuyển sang vị trí $i + 1$.

$$dp[i][1] = \max(dp[i][1], dp[i][0])$$
$$dp[i][2] = \max(dp[i][2], dp[i][1])$$

- Chuyển sang vị trí $i + 1$. Khi đó, ta sẽ tính $dp[i + 1][state_{mul}]$ dựa theo thuật toán Kadane và giá trị phần tử a_{i+1} (phụ thuộc vào trạng thái $state_{mul}$).

$$dp[i + 1][0] = \max(dp[i + 1][0], \max(0, dp[i][0]) + A[i])$$
$$dp[i + 1][1] = \max(dp[i + 1][1], \max(0, dp[i][1]) + A[i] * x)$$
$$dp[i + 1][2] = \max(dp[i + 1][2], \max(0, dp[i][2]) + A[i])$$

Kết quả cần tìm sẽ là $\max(dp[i][2])$ với $1 \le i \le n$

Độ phức tạp: $O(n)$ với n là số phần tử của dãy A.

Tag: DP