TRẠI HÈ HÙNG VƯƠNG
LẦN THỨ XV – SƠN LA

HƯỚNG DẪN GIẢI ĐỀ THI CHỌN HỌC SINH GIỎI
MÔN: TIN HỌC
Ngày thi: 27 tháng 7 năm 2019
Hướng dẫn này gồm có 07 trang

Bài 1. Ảnh đẹp

Được đề xuất bởi: Trường THPT Chuyên Tuyên Quang

Yêu cầu bài toán: Tìm đoạn con tổng lớn nhất có độ dài chẵn và độ dài ≥ 4.
Nhận xét: Giả sử đoạn tổng lớn nhất là đoạn $[u, v]$ ta có tổng

$$a_u + a_{u+1} + \cdots + a_v = S_v - S_{u-1}$$

Với $S_i = a_1 + a_2 + \cdots a + a_i$.
Đặt $p = u - 1$. Khi đó, với mỗi v, ta có giá trị S_p là giá trị nhỏ nhất trong tất cả các $S_p, S_{p-2}, S_{p-4}, S_{p-6}, \ldots$.
Giải thuật: Gọi f_i là giá trị nhỏ nhất trong tất cả các giá trị $S_i, S_{i-2}, S_{i-4}, S_{i-6}, \ldots$.
Dễ dàng tính $f_i = \min\,(f_{i-2}, S_i)$.
Từ đó, với mỗi giá trị v, đoạn con có tổng lớn nhất thỏa mãn có giá trị là $S_v - f_{v-4}$.
Độ phức tạp: $O(n)$

Bài 2. Đội hình thi đấu

Người đề xuất: Thầy Nguyễn Thanh Tùng

Giải thuật: Nguyên lý cực trị

Sắp xếp dữ liệu theo chiều tăng dần của cặp giá trị (b_i, a_i), **i** = 1 ÷ **n**.

Xét tiềm năng của đội **k** người. Gọi **x** là sức mạnh của đội trưởng và **y** – tổng độ dẻo dai của các người còn lại.

Dễ dàng thấy rằng chỉ có thể có 2 cách chọn:

- Chọn **k** người đầu tiên trong dãy đã sắp xếp,
- Chọn **k-1** người đầu tiên trong dãy đã sắp xếp và một người trong số **n-k** người (từ vị trí **k+1** đến **n**) còn lại làm đội trưởng.

Từ 2 cách chọn trên rút ra đội có tiềm năng nhỏ hơn.

Gọi **ia** là chỉ số nhỏ nhất thỏa mãn điều kiện $a_{ia} \le a_i$ với **i** = 1 ÷ **n** trong dãy đã sắp xếp.

Nếu **ia** > **k** thì đội cần chọn sẽ bao gồm **k-1** người đầu tiên trong dãy đã sắp xếp làm thành viên và người thứ **ia** làm đội trưởng.

Nếu **ia** ≤ **k** – đội cần chọn sẽ gồm **k** người đầu tiên. Vấn đề còn lại – xác định đội trưởng.

Gọi **s** là tổng độ dẻo dai của **k** người đầu tiên.

s = $\sum_{i=1}^{k} b_i$

d = $max\{b_i - a_i,\ i = 1 \div k\}$

Tiềm năng cần tìm sẽ là **s-d**.

Các giá trị **s** và **d** được cập nhật với chi phí O(1) khi chuyển từ **k** sang **k+1**.

Tổ chức dữ liệu

- Mảng `vector<pll> ba(n)` – lưu các cặp giá trị nguyên 64 bits (b_i, a_i),
- Mảng `vector<int64_t> f(n)` – lưu kết quả cần tìm.

Độ phức tạp của giải thuật: O(*n*log*n*).

Bài 3. Pha lê

Được đề xuất bởi: Trường THPT Chuyên Hạ Long và trường THPT Chuyên Thái Nguyên.

Giải thuật:

Xét tất cả các ước của n.

Một số kỹ thuật đếm số lượng phần tử khác nhau trong một đoạn:

- Sort đoạn tăng dần. Đếm số lượng cặp cạnh khác nhau.
- Sử dụng bảng đánh dấu. Đếm số lượng phần tử xuất hiện. Chú ý số lượng này chỉ thay đổi khi 1 số từ trạng thái số lần xuất hiện 0 ↔ 1
- Giả sử ta cần tính đoạn $[u, v]$. Gọi $tr[i]$ là vị trí bên trái gần i nhất có giá trị bằng a_i. Số phần tử khác nhau chính là số lượng p thỏa mãn $tr[p] < u$.

Lựa chọn kỹ thuật thứ 2 hoặc thứ 3 để đạt được độ phức tạp $O(n)$ với mỗi lần xét ước.

Tổng độ phức tạp $O(n \times k)$ với k là số ước của n.

Bài 4 (Bài 1 khối 11). Dãy chứa max

Được đề xuất bởi: Thầy Nguyễn Thanh Tùng

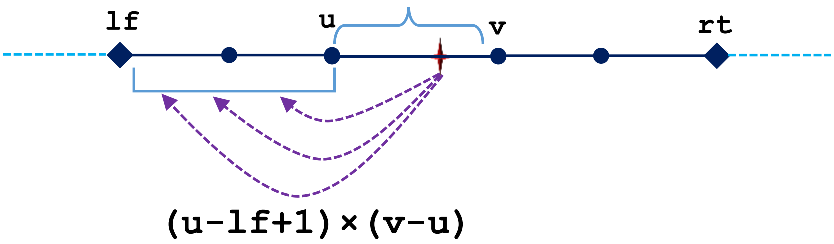
Giải thuật: Kỹ thuật 2 con trỏ và nguyên lý cực trị

Tạo hàng rào: $a_0 = a_{n+1} = 10^9+1$.

Lần lượt từ trái qua phải tìm đoạn [**lf**, **rt**] dài nhất các phần tử liên tiếp có giá trị không vượt quá **b**, Với mỗi đoạn [`lf`, `rt`]:

Duyệt a_i, `i` = `lf` ÷ `rt`: nạp các `i` có a_i = **b** vào hàng đợi `q`,

Với `u` và `v` – hai phần tử liên tiếp trong hàng đợi, `u` < `v`, `v` = `rt` nếu `u` là phần tử cuối cùng trong hàng đợi, số đoạn thỏa mãn điều kiện đầu bài và chứa phần tử a_u là `(u-lf+1)×(v-u)`.



Mỗi đoạn với `u` mới không trùng với các đoạn đã tính ở `u` cũ vì có chứa phần tử a_u mới, trong một đoạn mở [`u`, `v`) các đoạn được đếm khác nhau bởi điểm đầu hoặc điểm cuối hay cả hai.

Không có đoạn nào bị bỏ sót vì mỗi cực trị đều tham gia vào mọi đoạn chứa nó.

Tổ chức dữ liệu

- Mảng `vector<int> a` – chứa dữ liệu input,
- Hàng đợi `queue<int> q` – chứa tọa độ các cực trị trong một đoạn [`lf`, `rt`].

Độ phức tạp của giải thuật: O(*n*).

Bài 5 (Bài 3 – khối 11). Nhảy về đích

Được đề xuất bởi: Trường THPT Chuyên Bắc Giang

Giải thuật: Quy hoạch động

Gọi $dp[i][j]$ là số cách di chuyển tới ô i, j. Giả sử $x = c_{ij}$.

Giá trị dp_{ij} cộng tích lũy vào tới các ô $dp(i, j + 1), dp(i, j + 2), \dots, dp(i, j + x)$;

$$dp(i + 1, j), dp(i + 2, j), \dots dp(i + x, j)$$

Sub 4: Sử dụng update trên cây IT hoặc BIT để tính toán. Độ phức tạp $O(mnlog_2mn)$

Sub 5. Cập nhật in, out. Thay vì cập nhật đoạn dp_{ij}tăng trên $[u, v]$ trên cây IT, ta cập nhật tích lũy tại điểm u với giá trị tăng dp_{ij} và tại điểm $v + 1$ cập nhật giảm dp_{ij}.

Độ phức tạp $O(mn)$