

Trại Đông Đà Lạt 2022

dophanthuan@gmail.com

Khoa Học Máy Tính
Đại học Bách Khoa Hà Nội



Ngày 15 tháng 11 năm 2022

Bài 1. Bội chung nhỏ nhất

Bài 2. Chờ đợi

Bài 3. Mùa giáng sinh

Bội chung nhỏ nhất

Alice và Bob chơi một trò chơi gồm T lượt. Ở mỗi lượt chơi Alice chọn một số nguyên A , Bob chọn một số nguyên B . Nhiệm vụ của Alice phải đi tìm một số tự nhiên x không vượt quá B sao cho $\text{lcm}(x, A)$ lớn nhất có thể. Do giá trị A, B quá lớn nên Alice không thể tính nhanh được.

Yêu cầu: Cho các số nguyên dương A và B . Bạn hãy giúp Alice tìm ra giá trị $\text{lcm}(x, A)$ lớn nhất đó với $\text{lcm}(x, y)$ là bội chung nhỏ nhất của hai số x và y .

Subtask 1 ($A, B, T \leq 1000$): $O(B * T * \sqrt{A})$

Làm trực tiếp, thử từng giá trị x và tính $lcm(x, A)$.

Subtask 2 ($A, B \leq 10^9$; $T \leq 50000$): $O(T * \log A)$

- ▶ Tính $\gcd(x, A)$ với độ phức tạp $O(\log(A))$.
- ▶ Duyệt x từ B về 1, $res = \max(A * x / \gcd(x, A))$, dừng sớm khi $\gcd(x, A) = 1$.

Bài 1. Bội chung nhỏ nhất

Bài 2. Chờ đợi

Bài 3. Mùa giáng sinh

Chờ đợi

Vinh nhận được thông tin về n cửa hàng, cửa hàng thứ i ($1 \leq i \leq n$) sẽ mở cửa đón khách từ thời điểm L_i và không nhận khách từ thời điểm R_i trở đi. Có m du khách đã đặt lịch trình tham gia lễ hội. Du khách thứ j ($1 \leq j \leq m$) sẽ tới vào thời điểm k_j và dự kiến sẽ vào cửa hàng bất kỳ nào đó đang mở đón khách. Trường hợp không có cửa hàng nào mở, du khách sẽ đợi và vào cửa hàng mở cửa đón khách sớm nhất có thể.

Yêu cầu: Hãy giúp Vinh xác định thời gian chờ đợi của từng du khách để vào được cửa hàng. Biết thời gian di chuyển không đáng kể.

Subtask 1 (40% test có $m, n \leq 10^3$; $R_i \leq 10^6$

$\forall i = 1, 2, \dots, n$, và tại một thời điểm, không có 2 cửa hàng nào cùng mở cửa (hay với $i \neq j$, ta có $L_i > R_j$ hoặc $L_j > R_i$). $L_i < R_i \leq 10^6 \ \forall i = 1, 2, \dots, n$)

Duyệt trực tiếp từng cửa hàng.

Subtask 2 (30% test có $10^3 \leq m, n \leq 10^5$ và tại một thời điểm, không có 2 cửa hàng nào cùng mở cửa (hay với $i \neq j$, ta có $L_i > R_j$ hoặc $L_j > R_i$). $L_i < R_i \leq 10^6$ $\forall i = 1, 2, \dots, n$)

- ▶ Sắp xếp các cửa hàng theo chiều tăng dần thời gian mở cửa.
- ▶ Với mỗi truy vấn k , tìm kiếm nhị phân (lower_bound) tới cửa hàng gần nhất có thời điểm đóng cửa lớn hơn k .

Subtask 3 ($n, m \leq 10^5$)

Có 3 trường hợp:

- ▶ Không vào được cửa hàng nếu như $k > R_{max}$.
- ▶ Không phải chờ đợi nếu tồn tại 1 đoạn chứa k . Kiểm tra bằng cách tìm nhị phân đếm số điểm $R_i \geq k$ và số điểm $L_i > k$. Nếu 2 số lượng này khác nhau thì sẽ tồn tại đoạn chứa k .
- ▶ Trường hợp còn lại, thời gian đợi là khoảng từ k tới L_i gần nhất mà $L_i > k$.

Bài 1. Bội chung nhỏ nhất

Bài 2. Chờ đợi

Bài 3. Mùa giáng sinh

Mùa giáng sinh — Noel

Chính quyền thành phố Dresden đã mua N cây thông trang trí trên con đường lớn nhất của thành phố này. Họ sẽ tiến hành đặt N cây thông thành 2 hàng ở 2 bên của con đường. Một cách đặt sẽ khiến con đường lộng lẫy nếu như:

- ▶ Mỗi hàng đều phải có cây thông.
- ▶ Chênh lệch chiều cao giữa 2 cây thông liên tiếp trong cùng một hàng là như nhau và chiều cao của các cây sắp xếp theo thứ tự không giảm.

Yêu cầu: Hãy đề ra một phương án đặt các cây thông sao cho con đường trở nên lộng lẫy. Hoặc nếu không thể, hãy báo cáo lên chính quyền.

Subtask 1 ($N \leq 15$): $O(2^N)$

Thực hiện duyệt các dãy nhị phân, các vị trí có số 0 thì cây thông ở các vị trí đó sẽ cùng một hàng, tương tự với các vị trí có số 1. Sau khi có được các cây thông ở hai hàng, thực hiện sắp xếp không giảm và kiểm tra.

Subtask 2 ($N \leq 300$): $O(N^3)$

Thực hiện sắp xếp dãy h_1, h_2, \dots, h_N không giảm. Nhận thấy mỗi hàng cây đều phải là một dãy cấp số cộng. Ta có các trường hợp sau:

- ▶ Với $N \leq 4$ thì ta chia các cây vào 2 hàng sao cho mỗi hàng không quá 2 cây thông.
- ▶ Với $N > 4$ thì luôn tồn tại một hàng có nhiều hơn 2 cây thông, giả sử đó là hàng bên trái. Bởi vì hàng bên trái là một dãy cấp số cộng nên ta chỉ cần quan tâm 2 số đầu tiên của hàng đó. Ta duyệt 2 giá trị $h_i, h_j (i < j)$ là hai cây đầu tiên của hàng bên trái. Nếu $N - 2$ cây còn lại có thể tạo ra một dãy cấp số cộng thì thỏa mãn, nếu không thì ta thêm một cây có giá trị bằng $2 * h_j - h_i$ vào hàng bên trái và kiểm tra $N - 3$ cây còn lại có thỏa mãn hay không. Cứ như vậy, ta cứ tiến hành thêm 1 cây mới (nếu có tồn tại) vào hàng bên trái và kiểm tra các cây còn lại có thể tạo ra dãy cấp số cộng hay không.

Subtask 3 ($N \leq 10^5$ và luôn tồn tại một phương án sao cho số lượng cây thông ở 2 hàng là như nhau)

Thực hiện sắp xếp dãy h_1, h_2, \dots, h_N không giảm. Ta xét cách giải quyết với $N > 4$ và chỉ có 3 trường hợp sau:

- ▶ Hai số đầu tiên của hàng bên trái là h_1, h_2 .
- ▶ Hai số đầu tiên của hàng bên trái là h_1, h_3 .
- ▶ Hai số đầu tiên của hàng bên trái là h_2, h_3 .

Với mỗi trường hợp, ta sẽ tạo một dãy cấp số cộng gồm $N/2$ phần tử, và kiểm tra $N/2$ phần tử còn lại có tạo thành một dãy cấp số cộng hay không.

ĐPT: $O(N * 3)$.

Subtask 4 ($N \leq 10^5$): $O(N * 3 * \log N)$

Thực hiện sắp xếp dãy h_1, h_2, \dots, h_N không giảm. Ta thực hiện các trường hợp như ở Subtask 3, vấn đề của ta là phải kiểm tra rằng sau khi thêm 1 cây mới vào hàng bên trái thì các cây còn lại có thể tạo thành một dãy cấp số cộng hay không. Để giải quyết vấn đề này, ta sử dụng 2 multiset S và T . Multiset S để lưu chiều cao h_i của các cây còn lại, multiset T lưu chênh lệch chiều cao giữa 2 cây liên tiếp trong S .

Khi thêm một cây mới có chiều cao y vào hàng bên trái thì:

- ▶ Xóa 1 giá trị y trong multiset S .
- ▶ Đặt x là số liền trước y , z là số liền sau y trong multiset S sau đó tiến hành xóa 2 giá trị $y - x$ và $z - y$ trong T , đồng thời thêm giá trị $z - x$ vào multiset T . Nếu tất cả các giá trị trong T bằng nhau (giá trị bé nhất bằng giá trị lớn nhất) chứng tỏ các cây còn lại có thể tạo thành 1 dãy cấp số cộng.