

Đại học quốc gia TP.HCM
Trường đại học công nghệ thông tin



Ngành: Khoa học máy tính

Môn học: CS112.Q11.KHTN

Lời giải bài tập nhóm 4

Giảng viên:

Nguyễn Thanh Sơn

Sinh viên thực hiện:

24520031 - Hà Xuân Thiện

24521901 - Trần Quang Trường

Contents

1	Tóm tắt đề bài	2
2	Subtask 1 ($n, m \leq 10^3$)	2
2.1	Phương pháp thiết kế thuật toán sử dụng	2
2.2	Tại sao sử dụng	2
2.3	Sử dụng như thế nào trong bài toán	2
2.4	Phân tích độ phức tạp	2
2.4.1	Độ phức tạp thời gian	2
2.4.2	Độ phức tạp không gian	3
3	Subtask 2 ($W, H \leq 10^3$)	3
3.1	Phương pháp thiết kế thuật toán sử dụng	3
3.2	Tại sao sử dụng	3
3.3	Sử dụng như thế nào trong bài toán	3
3.4	Phân tích độ phức tạp	3
3.4.1	Độ phức tạp thời gian	3
3.4.2	Độ phức tạp không gian	4
4	Subtask 3 (Không ràng buộc bổ sung)	4
4.1	Phương pháp thiết kế thuật toán sử dụng	4
4.2	Tại sao sử dụng	4
4.3	Sử dụng như thế nào trong bài toán	4
4.4	Phân tích độ phức tạp	5
4.4.1	Độ phức tạp thời gian	5
4.4.2	Độ phức tạp không gian	5

1 Tóm tắt đề bài

Bản đồ là một mặt phẳng hình chữ nhật kích thước $W \times H$ ($W, H \leq 10^9$). Có $n \leq 10^5$ vùng theo dõi Aura (hình chữ nhật trục thẳng) không giao nhau và không chạm biên bản đồ. Có $m \leq 10^5$ Hunter tại vị trí (C_j, R_j) mang lượng Aura t_j .

Yêu cầu: Với mỗi Hunter, xác định vùng chứa nó (vì các vùng không giao nhau nên luôn duy nhất một vùng hoặc không vùng nào). In ra tổng Aura của từng vùng (theo thứ tự input) và cuối cùng là tổng Aura nằm ngoài mọi vùng.

2 Subtask 1 ($n, m \leq 10^3$)

2.1 Phương pháp thiết kế thuật toán sử dụng

Brute Force.

2.2 Tại sao sử dụng

Bài toán có đặc điểm quy mô nhỏ ($n, m \leq 10^3$), dẫn đến không gian tìm kiếm hữu hạn và có thể liệt kê hết các khả năng mà không vượt giới hạn thời gian. Đặc điểm này khớp với tính chất của Brute Force, vốn dành cho các vấn đề mà số lượng kết hợp có thể kiểm tra trực tiếp mà không cần tối ưu hóa phức tạp.

2.3 Sử dụng như thế nào trong bài toán

Với mỗi Hunter j (tổng m Hunter), duyệt qua tất cả n vùng để kiểm tra xem điểm (C_j, R_j) có nằm nghiêm ngặt bên trong vùng i (kiểm tra $X_i \leq C_j \leq A_i$ và $Y_i \leq R_j \leq B_i$). Nếu có, cộng t_j vào tổng của vùng i ; nếu không thuộc vùng nào, cộng vào phần outside.

2.4 Phân tích độ phức tạp

2.4.1 Độ phức tạp thời gian

- Ở bước tiền xử lý: Không cần tiền xử lý do đó độ phức tạp thời gian là $\Theta(1)$.

- Ở bước xử lý: Với mỗi Hunter cần duyệt n vùng, mỗi kiểm tra containment nên mất $O(n)$. Do đó, tổng m Hunter mất $O(nm)$ để kiểm tra.

\Rightarrow Do đó, độ phức tạp thời gian của bài toán là $< \Theta(1), \Theta(nm) >$.

2.4.2 Độ phức tạp không gian

Vì không có bước tạo dữ liệu nào nên độ phức tạp không gian của bài toán là $\Theta(1)$.

3 Subtask 2 ($W, H \leq 10^3$)

3.1 Phương pháp thiết kế thuật toán sử dụng

Transform and Conquer (Biến đổi để trị).

3.2 Tại sao sử dụng

Bài toán có đặc điểm bản đồ nhỏ ($W, H \leq 10^3$), cho phép tạo mảng 2D để precompute mà vẫn không vượt bộ nhớ. Việc biến đổi cách biểu diễn dữ liệu là đặc điểm của Transform and Conquer. Điều này giúp giảm thời gian truy vấn từ $\Theta(m)$ xuống $\Theta(1)$.

3.3 Sử dụng như thế nào trong bài toán

Tạo mảng 2D kích thước $W \times H$ để lưu lượng Aura tại từng ô, cộng t_j vào ô (C_j, R_j) . Sau đó tính prefix sum 2D để truy xuất tổng Aura trong bất kỳ hình chữ nhật con nào trong $O(1)$.

3.4 Phân tích độ phức tạp

3.4.1 Độ phức tạp thời gian

- Ở bước tiền xử lý: Bước xây dựng prefix sum 2D tốn độ phức tạp thời gian là $\Theta(m + WH)$.
- Ở bước xử lý: Vì đã có mảng tổng tiền tố nên thời gian để truy vấn với mỗi vùng là $\Theta(1)$. Do đó độ phức tạp thời gian là $\Theta(n)$.

\Rightarrow Do đó, độ phức tạp thời gian của bài toán là $< \Theta(m + WH), \Theta(n) >$.

3.4.2 Độ phức tạp không gian

- Ở bước tiền xử lý, ta cần xây dựng mảng prefix sum 2D nên tổn độ phức tạp không gian là $\Theta(WH)$.
- Ở bước xử lý, ta không tạo thêm gì hết nên độ phức tạp không gian là $\Theta(1)$.

\Rightarrow Do đó, độ phức tạp không gian của bài toán là $\Theta(WH)$.

4 Subtask 3 (Không ràng buộc bổ sung)

4.1 Phương pháp thiết kế thuật toán sử dụng

Transform-and-Conquer (Biến đổi để trị).

4.2 Tại sao sử dụng

Bài toán có đặc điểm ràng buộc lớn ($W, H \leq 10^9, n, m \leq 10^5$) khiến không thể dùng lưới được. Vì các vùng không giao nhau nên ta có ý tưởng tại mỗi Hunter tìm kiếm có tồn tại vùng nào chứa Hunter đó không. Thay vì biểu diễn trên bảng lưới 2D ta biểu diễn bài toán về 1D. Coi một chiều là trục thời gian, chiều còn lại ta biểu diễn thành các sự kiện (Sweep line). Vì bài toán có đặc điểm biến đổi cách biểu diễn dữ liệu nên ta sử dụng phương pháp thiết kế Transform and Conquer.

4.3 Sử dụng như thế nào trong bài toán

Ta coi mỗi vùng theo dõi là một khoảng trên trục từ X_i đến A_i trục x . Tại mỗi vị trí x , duy trì tập các khoảng dọc y đang active). Với mỗi Hunter tại $x = C_j$, kiểm tra xem R_j có nằm trong khoảng dọc nào của tập active không.

Cụ thể: Tại mỗi vị trí x , có thể có một trong 3 loại sự kiện sau:

- Sự kiện mở, với mỗi vùng i , tại $x = X_i$, ta thêm đoạn (Y_i, B_i) vào tập active.

- Sự kiện truy vấn, với mỗi Hunter j , tại $j = C_j$, ta kiểm tra xem R_j có thuộc khoảng nào trong tập active không.
- Sự kiện đóng, với mỗi vùng i , tại $X = A_i$, ta loại bỏ đoạn (Y_i, B_i) khỏi tập active.

Để biểu diễn tập sự kiện, ta sắp xếp tập sự kiện ưu tiên theo x trước rồi ưu tiên theo sự kiện mở, tới sự kiện truy vấn, rồi tới sự kiện đóng.

Để biểu diễn tập active, ta có thể sử dụng set để thực hiện thao tác thêm/xóa/tìm kiếm trong $\log n$. Cụ thể, ta lưu tập set gồm các khoảng (l, r) và chứa vị trí i của vùng (Nói cách khác một tuple (l, r, i)). Khi truy vấn một hunter có vị trí trên trục dọc là y thì chỉ cần tìm khoảng có l lớn nhất mà $l \leq y$, nếu $l \leq y \leq r$ thì tức đoạn i chứa hunter đó.

4.4 Phân tích độ phức tạp

4.4.1 Độ phức tạp thời gian

- Ở bước xây dựng tập sự kiện, ta mất độ phức tạp thời gian là $\Theta((n + m) \log(n + m))$.
- Ở bước duyệt tập sự kiện, với mỗi sự kiện mở/đóng/truy vấn đều mất $\log n$. Do đó, để duyệt tập sự kiện mất độ phức tạp thời gian là $\Theta((n + m) \log n)$.

Do đó, độ phức tạp thời gian của bài toán là

$$< \Theta((n + m) \log(n + m)), \Theta((n + m) \log n) >$$

4.4.2 Độ phức tạp không gian

- Để xây dựng tập sự kiện, ta mất độ phức tạp không gian là $\Theta(n + m)$.
- Tại bước duyệt tập sự kiện, ta cần duy trì một cái set, và một mảng lưu số hunter có trong một vùng do đó độ phức tạp không gian là $\Theta(n)$.

Do đó, độ phức tạp không gian của bài toán là $\Theta(n + m)$.