

ĐÁP ÁN

Môn thi: **TIN HỌC**

Ngày thi thứ hai: 20 tháng 10 năm 2020

Thời gian làm bài: 180 phút

Bài 5. Chia điểm (7 điểm)

Có thể kiểm tra vị trí tương đối của hai điểm với một đường thẳng bằng cách viết phương trình đường thẳng hoặc tính diện tích tam giác...

- **Sub 1+2:** cố định 1 điểm, for điểm còn lại, kiểm tra xem mỗi trường hợp có bao nhiêu điểm ở hai phía đường thẳng.
- **Sub 3:** Chọn 1 điểm có toạ độ y nhỏ nhất (thấp nhất), chọn 1 điểm có toạ độ x nhỏ nhất (trái nhất) để tính góc, sau đó sort lại các điểm theo góc. Hoặc cách khác dùng bao lồi.

Bài 6. Đoạn thẳng (7 điểm)

- Loại bỏ các đoạn thẳng $[x, y]$ bị "phủ" bởi các đoạn thẳng khác.
- Sắp xếp các đoạn thẳng $[a_i, b_i]$ còn lại theo a_i tăng dần.
- Với mỗi đoạn thẳng $[a_i, b_i]$, giả sử ta chọn đoạn này để "phủ" đoạn $[u, v]$ cần tìm, đoạn thẳng tiếp theo cần chọn để phủ đoạn $[u, v]$ là đoạn $[a_j, b_j]$ có $a_j \leq b_i$ và j lớn nhất. Đặt $\text{next}(i) = j$.
- Tính $t = \text{next}(\text{next}(\dots \text{next}(i) \dots))$ (k lần).
- Loại bỏ những đoạn $[a_i, b_i]$ bị "phủ" bởi các đoạn khác và sắp xếp chúng tăng dần theo a_i .
- Với mỗi truy vấn $[c, d]$, đoạn $[u, v]$ dài nhất cần tìm thuộc một trong 3 TH sau:
 - TH1: là giao của đoạn $[a_i, b_i]$ với đoạn $[c, d]$ trong đó a_i lớn nhất và $a_i \leq c$.
 - TH2: là giao của đoạn $[a_i, b_i]$ với đoạn $[c, d]$ trong đó b_i nhỏ nhất và $b_i \geq d$.
 - TH3: là đoạn $[a_i, b_i]$ mà $c \leq a_i \leq b_i \leq d$.

TH1 và TH2 có thể dùng chèn nhị phân để tìm, còn TH3 các đoạn $[a_i, b_i]$ cần xét sẽ nằm liên tiếp nhau nên đoạn có độ dài lớn nhất có thể dùng Segment Tree để xác định.

Bài 7. Tô màu (6 điểm)

- Sub 1: duyệt hoán vị m rồi kiểm tra
- Sub 2: coi mỗi đoạn tô là một đỉnh của đồ thị, xây dựng đồ thị. Bài toán có nghiệm khi đồ thị xây dựng phải là DAG (đồ thị có hướng không chu trình), nghiệm của bài toán chính là bài toán sắp xếp topo. Số cạnh của đồ thị tối đa sẽ là $N \times M$, do tại mỗi vị trí ta có tối đa M cạnh, thuật toán sắp xếp topo thực hiện trong $O(V + E)$ do đó độ phức tạp của sub này là $O(N \times M)$
- Sub 3: tương tự Sub 2 ta xây dựng đồ thị rồi thực hiện thuật toán sắp xếp topo, tuy nhiên để giảm số cạnh ta xây dựng một cây Segment tree.
- Sub 4: Ta có nhận xét sau
 - Một đoạn tô đoạn $[l, r]$ sẽ được thực hiện sau các đoạn khác nếu tất cả các ô trong đoạn $[l, r]$ có màu trùng với màu của đoạn đó
 - Dựa vào nhận xét trên ta có thuật toán tham sau: ta đi tìm lần lượt các đoạn theo thứ tự từ cuối về đầu.
 - Với thuật toán này độ phức tạp sẽ là $O(NM)$
- Sub 5: Với thuật toán của sub 4 ta thấy việc mất thời gian nhất là tìm đoạn mà các ô chưa bị đánh dấu có màu trùng mới màu của đoạn đó, với sub 5 này ta cải tiến thuật toán trên bằng disjoint set
- Sub 6: Vẫn tư tưởng như sub 4 ta xây dựng một cây segment tree như sau để tìm lần lượt các đoạn tô:
 - Với mỗi node $[l, r]$ ta lưu lại màu min và max của các ô chưa được đánh dấu, và danh sách các đoạn tô $[s, t]$ mà $s \leq l \leq r \leq t$
 - Với mỗi đoạn tô ta lưu d_i là số node của cây segment tree lưu đoạn tô i mà màu của các ô của các node này vẫn chưa còn đúng màu của đoạn tô i
 - Giả sử ta tìm được một đoạn tô i $[l, r]$ thì cập nhật lại các node trên cây segment tree có chứa đoạn tô i này, ta cập nhật lại màu min và max của node đó,
 - Nếu màu min và max là giống nhau có nghĩa là các ô trong đoạn $[l, r]$ chưa được đánh dấu có màu giống nhau, ta sẽ giảm d_i của đoạn tô i mà màu của đoạn tô i này trùng với màu min, nếu d_i bằng 0 thì ta biết rằng đoạn tô i này chính là đoạn tiếp theo cần tìm

----- Hết -----