



ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Thực hành TTUĐ

Week 3

Chia đề trị, Tham lam

Bài 1. Disjoint Segment

Tìm tập các đoạn không giao nhau lớn nhất

- Cho đầu vào là tập X gồm n đoạn $X = \{(a_1, b_1), \dots, (a_n, b_n)\}$ với $a_i < b_i$
- Hãy tìm và đưa ra tập con lớn nhất của X chứa các đoạn này sao cho không có 2 đoạn nào giao nhau

Input

- Dòng 1 là giá trị của n ($1 \leq n \leq 100000$)
- Các dòng tiếp theo là các khoảng thời gian (a_i, b_i) ($1 \leq a_i \leq b_i \leq 1000000$)

Output

- Số lượng phần tử của tập con lớn nhất

- Example

stdin	stdout
6 0 10 3 7 6 14 9 11 12 15 17 19	4

Disjoint Segment- Hint

- Áp dụng tham lam để chọn được số lượng đoạn lớn nhất
- Nếu có 2 đoạn trùng nhau, ta sẽ ưu tiên chọn khoảng có điểm cuối nhỏ hơn
- Sắp xếp các đoạn tăng dần theo điểm kết thúc
- Biến phụ trợ:
 - last**: lưu điểm kết thúc của đoạn trước đó ta đã chọn
- Duyệt qua tất cả các đoạn, kiểm tra xem điểm bắt đầu của điểm đó có lớn hơn điểm kết thúc của điểm cuối cùng ta đã chọn hay không
- Nếu đoạn đang xét thỏa mãn, tăng số lượng đoạn được chọn lên, và cập nhật last

Bài 2 MAX-DISTANCE SUB-SEQUENCE

- Cho 1 tập gồm N điểm trong không gian 1 chiều (đường thẳng).
- Độ kết nối của cụm (tập con) gồm C phần tử được tạo ra từ tập ban đầu được tính bằng khoảng cách của 2 điểm gần nhau nhất trong cụm
- VD. Cụm gồm $C=5$ điểm là 12,30,15,10,37 thì độ kết nối của cụm sẽ là 2, chính là khoảng cách nhỏ nhất giữa 12 và 10

MAX-DISTANCE SUB-SEQUENCE

- Example

stdin	stdout
1 5 3 1 2 8 4 9	3

Dòng đầu tiên sẽ là số lượng test trong bộ test T ($1 \leq T \leq 20$)

- Các dòng tiếp theo sẽ là chi tiết các test. Mỗi test tương ứng gồm 2 giá trị là N và C, trong đó N là số lượng phần tử và C là kích thước tập con. Trong 1 bộ dữ liệu thì giá trị N sẽ là giống nhau trong các test.
- Các dòng tiếp sau sẽ lần lượt là giá trị tọa độ của các điểm trong tập N điểm
- Giá trị N ($2 \leq N \leq 100,000$) và tọa độ các điểm x_1, \dots, x_n ($0 \leq x \leq 1,000,000,000$)
- Đầu ra sẽ là độ kết nối lớn nhất của tập con với kích thước C tương ứng (với các test). Nếu có T test thì sẽ có T giá trị đầu ra, với mỗi kết quả đầu ra được in trên 1 dòng.

- Sắp xếp lại mảng theo thứ tự tăng dần
- Sử dụng thuật toán tìm kiếm nhị phân để thử kết quả
- Thực hiện tìm kiếm nhị phân để tìm kết quả tốt nhất d:
 - $0 \leq d \leq (a[n] - a[1])$
- Bắt đầu thử từ giá trị $d = \text{mid}$:
- Kiểm tra xem có thể chọn được ít nhất c phần tử, mỗi phần tử cách nhau ít nhất d đơn vị hay không
- Nếu giá trị $d = \text{mid}$ là 1 giá trị khả thi, ta tiếp tục tìm kiếm ở nửa bên phải.
- Ngược lại, ta sẽ tìm kiếm tiếp ở nửa bên trái

Bài 3 Inversion

- Cho một dãy số nguyên a_1, a_2, \dots, a_n . Một cặp (a_i, a_j) được gọi là bị đảo ngược nếu $i < j$ nhưng giá trị $a_i > a_j$
- Hãy đếm số lượng các cặp bị đảo ngược trong dãy

Input

- Dòng 1: là giá trị của n ($1 \leq n \leq 10^6$)
- Dòng 2: là các giá trị của a_1, a_2, \dots, a_n . ($0 \leq a_i \leq 10^6$)

Output

Số lượng cặp bị đảo Q module $10^9 + 7$

stdin	stdout
6 3 2 4 5 6 1	6

Inversion: Hint

- Ý tưởng tương tự như sắp xếp hợp nhất (merge sort), chia mảng thành hai nửa (gần) bằng nhau trong mỗi bước cho đến khi đạt được trường hợp cơ sở (trường hợp khi chỉ có 1 phần tử trong mảng)
- Tạo một hàm đệ quy để chia mảng thành hai nửa.
 - Kết quả của mảng con từ left đến right là **tổng** số lần đảo ngược trong nửa đầu, số lần đảo ngược trong nửa sau và số lần đảo ngược khi hợp nhất cả hai.
- Tạo một hàm hợp nhất đếm số lần nghịch đảo khi hai nửa của mảng được hợp nhất
 - Nếu $a[i] > a[j]$, thì có **(mid – i) nghịch đảo** (với $a[i]$ thuộc nửa mảng thứ nhất, $a[j]$ thuộc nửa mảng thứ 2, mid là phần tử cuối cùng của nửa thứ nhất)
 - Bởi vì mảng con bên trái và bên phải đã được sắp xếp, nên tất cả các phần tử còn lại trong mảng con bên trái ($a[i+1], a[i+2] \dots a[mid]$) sẽ lớn hơn $a[j]$