

KHẢO SÁT ĐỘI TUYỂN HSG CẤP TỈNH

Có 4 bài, mỗi bài 5 điểm

Câu 1. SIGHTS.cpp

Trên con đường chinh phục đỉnh Fansipan (nóc nhà của Đông Nam Á cao 3143m), có N điểm tham quan, có view cực đẹp và có thể check-in rất ngẫu. Các vị trí đánh số từ 1 đến N , điểm thứ i nằm cách vị trí chân núi i mét, có độ đẹp là b_i . Tí muốn chọn một đoạn $[L, R]$ và check-in tại 3 vị trí khác nhau trong đó là i_1, i_2, i_3 , nếu di chuyển càng xa thì càng mệt, nên độ hài lòng đoạn $[L, R]$ là: $b_{i_1} + b_{i_2} + b_{i_3} - (R - L)$.

Hỏi độ hài lòng lớn nhất của một đoạn bất kì là bao nhiêu?

Input: Dòng đầu tiên ghi số N là độ dài đoạn đường ($3 \leq N \leq 5 \cdot 10^5$)

Dòng tiếp theo ghi độ đẹp của từng đoạn là $b_1, b_2, b_3 \dots b_N$ ($1 \leq b_i \leq 10^8$)

Output: Ghi ra độ đẹp lớn nhất của một đoạn bất kì.

Ví dụ:

SIGHTS.inp	SIGHTS.out	Giải thích
5 5 1 4 2 3	8	Chọn cả đoạn, độ hài lòng là: $5 + 4 + 3 - (5 - 1) = 8$
4 1 1 1 1	1	Chọn bất kì một đoạn nào đủ 3 điểm, chẳng hạn chọn đoạn $[1,3]$, độ hài lòng là: $1 + 1 + 1 - (3 - 1) = 1$

Subtask 1: 30% test có $N \leq 500$;

Subtask 2: 30% test có $N < 10000$;

Subtask 3: 40% test có $N \leq 5 \cdot 10^5$.

Câu 2. Kone2.cpp

Một xâu gọi là xâu nhị phân nếu chỉ chứa hai ký tự “0” hoặc “1”. Xâu v gọi là xâu con của S nếu xâu v có độ dài khác 0 và gồm các ký tự liên tiếp trong xâu S . Ví dụ: xâu “010” có các xâu con là “0”, “1”, “0”, “01”, “10”, “010”.

Yêu cầu: Cho trước một giá trị K , hãy đếm xem có bao nhiêu cách chọn chỉ số (i, j) để tạo ra một xâu con, mà số số lượng ký tự 1 hơn số lượng ký tự 0 đúng K .

Dữ liệu vào: Từ tệp văn bản Kone2.INP có cấu trúc:

Dòng 1 chứa một số nguyên K ($0 \leq K \leq 10^6$);

Dòng 2 chứa một xâu nhị phân có độ dài không quá 10^6 .

Kết quả ra: Ghi ra tệp văn bản Kone2.OUT

Một số nguyên duy nhất là kết quả tìm được.

Ví dụ:

Kone2.INP	Kone2.OUT	Giải thích
1 01010	3	Có 3 cách chọn chỉ số là $\{2,2\}, \{2,4\}, \{4,4\}$ để số lượng ký tự 1 hơn ký tự 0 đúng K

Có 50% test có $N \leq 1000$; Có 25% test có $N \leq 10^4$; Có 25% test có $N \leq 10^5$.

Câu 3. M2019.cpp

Cho một chuỗi S chỉ chứa các số từ 1 đến 9. Gọi $|S|$ là độ dài của chuỗi. Hãy đếm số lượng cặp chỉ số (i, j) mà $1 \leq i \leq j \leq |S| \leq 200000$ sao cho khi viết các kí tự từ i đến j tạo thành một số tự nhiên chia hết cho 2019.

Dữ liệu vào: Cho chuỗi S chỉ chứa các số từ 1 đến 9.

Kết quả ra: Ghi ra số cặp (i, j) sao cho tạo thành số chia hết cho 2019

Ví dụ:

M2019.inp	M2019.out	Giải thích
1817181712114	3	Có thể chọn 3 cặp là (1,5), (5,9), (9,13)
14282668646	2	Có thể chọn 2 cặp là: (3,7), (7,11)
2119	0	Không có cặp nào

Subtask 1: 30% test có $|S| \leq 500$;

Subtask 2: 30% test có $|S| \leq 50000$;

Subtask 3: 40% test có $|S| \leq 200000$.

Câu 4. SPLIT.cpp

Famer John có N con bò, mỗi con được đặt tại các vị trí khác nhau trên cánh đồng là mặt bằng toạ độ hai chiều Oxy . John muốn bảo vệ đàn bò bằng cách quây chúng lại bằng hình chữ nhật có cạnh song song với trục Ox, Oy . Ông ta muốn diện tích vùng bảo vệ càng nhỏ càng tốt, do doanh thu từ sữa gần đây giảm mạnh, nên John cần phải cắt giảm chi phí bảo trì vùng bảo vệ này bằng cách sử dụng hai vùng nhỏ hơn để bảo vệ được đàn bò của mình, hai vùng vẫn có cạnh song song với trục toạ độ, không được giao nhau, không được tiếp xúc nhau. Hỏi diện tích vùng bảo vệ đàn bò của John giảm được nhiều nhất là bao nhiêu?

Các con bò biên của hình chữ nhật vẫn được tính là được bảo vệ, vùng bảo vệ có thể có diện tích là 0 (cạnh là 0).

Dữ liệu vào: SPLIT.inp

Dòng đầu ghi số N ($3 \leq N \leq 50000$) là số con bò của John

N dòng tiếp theo, mỗi dòng ghi 2 số là toạ độ của các con bò, dòng thứ i ghi (x_i, y_i) là toạ độ con bò thứ i , biết $1 \leq x_i, y_i \leq 10^9$.

Kết quả ra: SPLIT.out

Là diện tích bảo vệ được giảm nhiều nhất khi sử dụng 2 vùng bảo vệ

Ví dụ:

SPLIT.inp	SPLIT.out	Giải thích
6 4 2 8 10 1 1 9 12 14 7 2 3	107	Diện tích ban đầu sử dụng một vùng bảo vệ: 143. Diện tích lúc sau, sử dụng hai vùng bảo vệ: 36. Vùng 1 là các con bò: 1,3,6 Vùng 2 là các con bò: 2,4,5

Subtask 1: 50% test có $N \leq 500$;

Subtask 2: 50% còn lại

