



ĐỀ CHÍNH THỨC

Ngày thi: 04 tháng 8 năm 2023

Thời gian làm bài: 180 phút (không kể thời gian giao đề)

Đề thi gồm có 03 trang

Trại hè Hùng Vương Confessions - fb.com/cfs.traihehungvuong

Họ và tên thí sinh: Số báo danh:

TỔNG QUAN ĐỀ THI

Bài	Tên bài	Tệp chương trình	Tệp dữ liệu	Tệp kết quả	Điểm
1	BẮN BI	MARBLES.*	MARBLES.INP	MARBLES.OUT	100
2	TẮC ĐƯỜNG	JAMONE.*	JAMONE.INP	JAMONE.OUT	100
3	ĐẾM DÂY	CSA.*	CSA.INP	CSA.OUT	100

Dấu * được thay thế bởi *pas* hoặc *cpp* của ngôn ngữ lập trình sử dụng tương ứng là Pascal hoặc C++.

BÀI 1. BẮN BI

Trường của Vinh đang tổ chức một giải bắn bi với luật chơi khá đặc biệt. Ban đầu có n viên bi được đánh số $1, 2, 3, \dots, n$ từ trái sang phải, viên bi ở vị trí i có hệ số điểm là a_i . Mỗi lần bắn, nếu bắn trúng một viên bi ở vị trí j , người chơi sẽ được số điểm bằng tích hệ số điểm của viên bi đó với tổng hệ số điểm của 2 viên bi bên cạnh. Sau khi bắn, viên bi ở vị trí j sẽ bị loại bỏ và các viên bi bên phải (vị trí $j + 1, j + 2, \dots$) sẽ được đẩy sang trái một vị trí.

[Trại hè Hùng Vương Confessions - fb.com/cfs.traihehungvuong](https://fb.com/cfs.traihehungvuong)

Ví dụ: Dây bi ban đầu là $\{2, 5, 7, 3, 8, 1\}$.

- Khi bắn vào viên bi vị trí 3, người chơi được $7 \times (5 + 3) = 56$ điểm, dây bi trở thành $\{2, 5, 3, 8, 1\}$
- Khi bắn tiếp vào viên bi vị trí 4, người chơi được $8 \times (3 + 1) = 32$ điểm, dây bi trở thành $\{2, 5, 3, 1\}$

Mỗi người cần thực hiện $n - 2$ lần bắn, đồng thời không được bắn vào một trong hai vị trí đầu tiên và cuối cùng.

Lần này, khi đến với hội thi, Vinh chưa ôn tập thuật toán bắn tối ưu nên Vinh đã chọn một cách bắn ngẫu nhiên, xác định bởi dãy $b_1, b_2, b_3, \dots, b_{n-2}$. Lượt bắn thứ i ($1 \leq i \leq n - 2$), Vinh sẽ bắn vào viên bi ở vị trí thứ b_i trong hàng hiện tại.

Yêu cầu: Hãy lập trình xác định tổng số điểm Vinh đạt được sau $n - 2$ lượt bắn.

Dữ liệu: Vào từ file **MARBLES.INP**

- Dòng đầu chứa số nguyên dương n ($3 \leq n \leq 2 \times 10^5$);
- Dòng thứ hai chứa n số nguyên dương a_1, a_2, \dots, a_n ($a_i \leq 10^5 \forall i = 1, 2, \dots, n$);
- Dòng thứ ba chứa $n - 2$ số nguyên b_1, b_2, \dots, b_{n-2} ($1 < b_i \leq n - i \forall i = 1, 2, 3, \dots, n - 2$).

Kết quả: Ghi ra file **MARBLES.OUT** một số nguyên duy nhất là tổng số điểm Vinh nhận được sau khi thực hiện $n - 2$ lượt bắn.

MARBLES . INP	MARBLES . OUT
6 2 5 7 3 8 1 3 4 3 2	121

Ràng buộc:

Trại hè Hùng Vương Confessions - fb.com/cfs.traihehungvuong

- Có 10% số test tương ứng 10% số điểm có $n \leq 1000$; $a_i = 1 \forall i = 1, 2, \dots, n$;
- Có 15% test khác tương ứng 15% số điểm có $n \leq 1000$; $b_j = 2 \forall i = 1, 2, \dots, n - 2$;
- Có 35% test khác tương ứng 35% số điểm có $n \leq 1000$;
- 40% số test còn lại có $n \leq 2 \times 10^5$.

BÀI 2. TẮC ĐƯỜNG

VP là một thành phố vô cùng đặc biệt. Ngoài vẻ đẹp tự nhiên thu hút khách du lịch, hệ thống giao thông cũng vô cùng đặc biệt. Thành phố được kết nối bởi n điểm với m con đường phục vụ di chuyển đảm bảo liên thông toàn thành phố. Con đường thứ i kết nối hai chiều giữa hai điểm u_i và v_i với thời gian di chuyển là p_i . Tuy nhiên, trong trường hợp xấu xảy ra tắc đường, để đi hết con đường này cần thời gian là q_i .

Hà là một quản lý trong một công ty vận tải, Hà đang thiết kế lộ trình cố định cho tuyến xe phục vụ đưa đón khách du lịch bắt đầu từ s , di chuyển qua một số con đường và kết thúc tại điểm t . Giả thiết trong trường hợp xấu nhất, chỉ có một con đường trên tuyến đường di chuyển đó gặp sự cố tắc đường.

Yêu cầu: Cho k truy vấn, mỗi truy vấn gồm 2 số $s, t (1 \leq s, t \leq n)$. Với mỗi truy vấn, hãy giúp Hà lựa chọn một lộ trình cố định sao cho trong trường hợp xấu, tổng thời gian di chuyển của xe là nhỏ nhất.

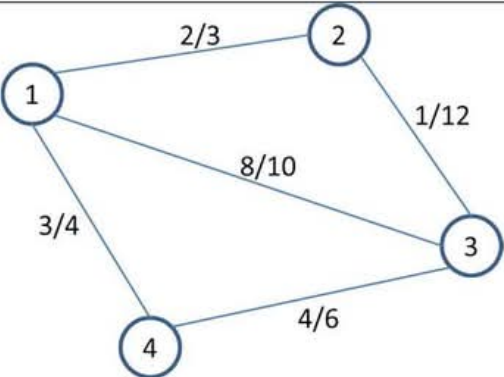
Dữ liệu: Vào từ file **JAMONE.INP**

- Dòng đầu chứa 3 số nguyên n, m, k ;
- m dòng tiếp theo, dòng thứ i chứa 4 số nguyên dương $u_i, v_i, p_i, q_i (u_i, v_i \leq n; p_i < q_i \leq 10^5 \forall i = 1, 2, \dots, m)$;
- k dòng cuối cùng, dòng thứ j chứa hai số nguyên dương s, t xác định thông tin trong truy vấn thứ $j (s, t \leq n; \forall j = 1, 2, \dots, k)$.

Trại hè Hùng Vương Confessions - fb.com/cfs.traihehungvuong

Kết quả: Ghi ra file **JAMONE.OUT** gồm k dòng, dòng thứ j đưa ra câu trả lời cho truy vấn thứ j là tổng thời gian di chuyển theo tuyến đường Hà lựa chọn trong trường hợp xấu nhất.

Ví dụ:

JAMONE . INP	JAMONE . OUT	MINH HỌA
4 5 3 1 2 2 3 1 3 8 10 1 4 3 4 3 4 4 6 2 3 1 12 1 3 2 4 2 3	9 6 11	

Giải thích:

- Để đi chuyển từ 1 tới 3 ta có 3 cách đi.
 - o Với cách đi $1 \rightarrow 3$ trong trường hợp xảy ra tắc thì thời gian đi chuyển là 10.
 - o Với cách đi $1 \rightarrow 2 \rightarrow 3$ trong trường hợp xảy ra tắc ở đường $1 \rightarrow 2$ thì thời gian đi chuyển là $3 + 1 = 4$ nhưng trong trường hợp xấu tắc ở $2 \rightarrow 3$ thì thời gian đi chuyển là $2 + 12 = 14$.
 - o Với cách đi chuyển $1 \rightarrow 4 \rightarrow 3$, trong trường hợp xảy ra tắc ở đường $1 \rightarrow 4$ thì thời gian đi chuyển là $4 + 4 = 8$, trong trường hợp xấu tắc ở $4 \rightarrow 3$ thì thời gian đi chuyển là $3 + 6 = 9$.

Ràng buộc:

Trại hè Hùng Vương Confessions - fb.com/cfs.traihungvuong

- 20% số test tương ứng 20% số điểm có $n \leq 10; m \leq 20; k = 1$
- 20% số test khác tương ứng 20% số điểm có $n \leq 100; m \leq 1000; k \leq 10$; và

$$q_1 - p_1 = q_2 - p_2 = q_3 - p_3 = \dots = q_m - p_m$$
- 30% số test khác tương ứng 30% số điểm có $n \leq 300; m \leq 1000; k \leq 10$
- 30% số test còn lại có $n \leq 1000; m \leq 5000; k \leq 10$.

BÀI 3. ĐẾM DÃY

Cho 3 số nguyên dương N, K, M và M bộ tham số: $(l_1, r_1, v_1), (l_2, r_2, v_2), \dots, (l_M, r_M, v_M)$.

Đếm số lượng dãy A khác nhau, gồm N phần tử số nguyên thỏa mãn:

- $0 \leq A[j] < 2^K$ với mọi $1 \leq j \leq N$.
- Với mọi $1 \leq i \leq M$ thì $A[l_i] \& A[l_i + 1] \& A[l_i + 2] \& \dots \& A[r_i] = v_i$.
Trong đó $\&$ là kí hiệu toán tử *AND*.

Dữ liệu: Vào từ file **CSA.INP**

- Dòng đầu tiên gồm ba số nguyên N, K, M ($N \leq 10^5, K \leq 30, M \leq 10^5$);
- M dòng tiếp theo, dòng thứ i là bộ ba số l_i, r_i , và v_i ($1 \leq l_i \leq r_i \leq N; 0 \leq v_i < 2^K$).

Kết quả: Ghi ra file **CSA.OUT**

Gồm một dòng duy nhất là kết quả bài toán khi lấy phần dư khi chia cho 1000000007. Hai dãy A và B được coi là khác nhau nếu tồn tại một vị trí i ($1 \leq i \leq N$) mà $A_i \neq B_i$.

Ví dụ:

CSA.INP	CSA.OUT
4 2 2 1 3 2 3 4 1	3

Giải thích: có 3 dãy thỏa mãn là $\{2, 2, 3, 1\}, \{2, 3, 3, 1\}, \{3, 2, 3, 1\}$

Ràng buộc:

- 15% số test tương ứng 15% số điểm có $M, N \leq 10; K \leq 2$;
- 15% số test khác tương ứng 15% số điểm có $M, N \leq 1000$ và $\forall i: 1 \leq i < m$ thì $r_i < l_{i+1}$;
- 20% số test khác tương ứng 20% số điểm có $K = 1; N, M \leq 1000$;
- 20% số test khác tương ứng 20% số điểm có $M, N \leq 1000$;
- 30% số test còn lại không có điều kiện gì thêm.

----- HẾT -----

*Thí sinh không sử dụng tài liệu.
Cán bộ coi thi không giải thích gì thêm.*



HƯỚNG DẪN GIẢI

BÀI 1: BẮN BI

Subtask 1: $n \leq 1000$; $a_i = 1 \forall i = 1, 2, \dots, n$

Mỗi lần bắn được $1 \times (1 + 1) = 2$ điểm. Vì vậy kết quả là $2 \times (n - 2)$

Subtask 2: $n \leq 1000$; $b_j = 2 \forall j = 1, 2, \dots, n - 2$

- Tất cả các lần bắn đều bắn vào vị trí 2.
- Vậy kết quả là: $a_2 \times (a_1 + a_3) + a_3 \times (a_1 + a_4) + \dots + a_{n-1}(a_1 + a_n)$
- Ta sử dụng một vòng lặp là tính được kết quả.

Subtask 3: $n \leq 1000$ [Trại hè Hùng Vương Confessions - fb.com/cfs.traihehungvuong](https://fb.com/cfs.traihehungvuong)

- Ta thực hiện đúng theo thao tác của mỗi lần bắn:
 - + Tính điểm cho lần bắn đó.
 - + Tính tiền các phần tử đứng sau vị trí bắn lên 1 đơn vị
- Độ phức tạp là $O(n^2)$

Subtask 4: $n \leq 2 \cdot 10^5$

- Dùng con trỏ móc nối hai đầu với mỗi viên bi.
- Để xác định vị trí bắn mới ta sử dụng cấu trúc Fenwick Tree hoặc Segment Tree.

BÀI 2: TẮC ĐƯỜNG

Subtask 1: $n \leq 10$; $m \leq 20$; $k = 1$

- Duyệt $i = 1 \dots m$ là cạnh xảy ra tắc đường.
- Ta xây dựng đồ thị mới với trọng số cạnh thứ j bằng $p_j \forall j \neq i$; riêng cạnh thứ i có trọng số là q_i .
- Tìm đường đi ngắn nhất từ s tới t .

Subtask 2: $n \leq 100$; $m \leq 1000$; $k \leq 10$; $q_1 - p_1 = q_2 - p_2 = q_3 - p_3 = \dots = q_m - p_m$

- Dùng thuật toán Floyd để tìm độ dài đường đi ngắn nhất giữa mọi cặp đỉnh với trọng số cạnh thứ i là p_i .
- Với mỗi truy vấn ta chỉ việc cộng thêm một đại lượng không đổi $q_1 - p_1$

Subtask 3: $n \leq 100$; $m \leq 1000$; $k \leq 10$

- Duyệt các cạnh theo thứ tự tăng dần về chênh lệch $q_i - p_i$

- Ta xây dựng đồ thị bằng cách thêm dần các cạnh theo thứ tự chèn lệch, khi đó kết quả sẽ là đường đi ngắn nhất cộng với chèn lệch (q-p) của cạnh vừa thêm. Sử dụng thuật toán Dijkstra để cài đặt đường đi ngắn nhất. ĐPT $O(k \times m^2 \log_2 n)$

Subtask 4: $n \leq 1000; m \leq 5000; k \leq 10$

- Thực hiện như subtask 3, nhưng mỗi khi ta thêm cạnh vào, không thực hiện tìm đường đi ngắn nhất lại từ đầu. Ta chỉ thực DFS cập nhật tiếp theo những giá trị cập nhật được theo cạnh vừa thêm hoặc cài đặt như thuật toán Bellman-Ford. ĐPT $O(m \times (m + n))$

BÀI 3: ĐẾM DÂY

Trại hè Hùng Vương Confessions - fb.com/cfs.traihehungvuong

Ta sẽ đếm số cách chọn các bit riêng biệt xong nhân kết quả lại với nhau.

Nhận xét, nếu một vị trí trong dãy nằm được phủ bởi một đoạn li, ri có giá trị 0 thì vị trí đó chắc chắn phải mang giá trị 0. Với các đoạn có giá trị 1 thì trong đoạn đó phải tồn tại ít nhất một vị trí mang giá trị 1.

Subtask 1: $M, N \leq 10; K \leq 2$: Duyệt vét cạn các dãy số.

Subtask 2: $M, N \leq 1000$ và với 2 số i, j ($1 \leq i < j \leq M$), ta có $r_i < l_j$:

Các đoạn điều kiện tách biệt nhau nên nếu một vị trí không nằm trong đoạn nào thì sẽ có 2 cách chọn, nếu nằm trong đoạn mang giá trị 1 thì chỉ có 1 cách chọn duy nhất, với đoạn mang giá trị 0 độ dài x , số cách chọn cho cả đoạn đó là $2^x - 1$.

Subtask 4: $M, N \leq 1000$

Trại hè Hùng Vương Confessions - fb.com/cfs.traihehungvuong

Quy hoạch động: $dp[i]$ là số dãy khác nhau mà vị trí 0 cuối cùng là i và tất cả các đoạn mang giá trị 0 ở đằng trước đều có ít nhất 1 số 0 trong đó. Ta có: $dp[0] = 1$

- Nếu i nằm trong một đoạn mang giá trị 1 thì $dp[i] = 0$
- Còn lại $dp[i] = \sum_{j=p}^{i-1} dp[j]$ trong đó p là số nguyên không âm, nhỏ nhất thỏa mãn không tồn tại chỉ số h sao cho $p < l_h \leq r_h < i$ và đoạn h mang giá trị 0, bởi vì khi đó đoạn h sẽ không thể có toàn số 1. Để tính số p ta có thể thêm dần các đoạn mang giá trị 1 và có cận phải $r < i$ và lấy $\max(l) + 1$.

Kết quả số cách chọn mỗi bit sẽ là $\sum_{j=p}^n dp[j]$

Độ phức tạp: $O(N^2 + M)$

Subtask 5: $N \leq 2 \cdot 10^5, K \leq 30, M \leq 2 \cdot 10^5$

Tối ưu hàm dp trong subtask 4 bằng tổng tiền tố và tính trước giá trị p là số nguyên không âm, nhỏ nhất thỏa mãn không tồn tại chỉ số h sao cho $p < l_h \leq r_h < i$ và đoạn h mang giá trị 1 với mỗi vị trí i . Độ phức tạp: $O(N + M)$