BỘ GIÁO DỤC VÀ ĐÀO TẠO	KỲ THI CHỌN HỌC SINH GIỎI QUỐC GIA TRUNG HỌC PHỔ THÔNG NĂM HỌC 2024 - 2025
ĐỀ THI CHÍNH THỨC	Môn: TIN HỌC Thời gian: 180 phút (*không kể thời gian giao đề*) Ngày thi thứ nhất: 25/12/2024 *Đề thi gồm 06 trang, 03 bài*

TỔNG QUAN ĐỀ THI

	Tiêu đề	File chương trình	File dữ liệu	File kết quả
Bài 1	Người giao hàng	SHIP.*	SHIP.INP	SHIP.OUT
Bài 2	Lập trình Robot AI	ROAI.*	ROAI.INP	ROAI.OUT
Bài 3	Số nguyên dương	PNUM.*	PNUM.INP	PNUM.OUT

- Dấu * được thay thế bởi PAS hoặc CPP hoặc PY tương ứng với ngôn ngữ lập trình `Pascal` hoặc `C++` hoặc `Python`.
- Mỗi bài bao gồm nhiều subtask, mỗi subtask bao gồm nhiều test đơn, điểm của thí sinh được tính theo từng test đơn.

Hãy lập trình giải các bài toán sau:

Bài 1. Người giao hàng (7,0 điểm)

Ở một quốc gia nọ có N hòn đảo, được đánh số lần lượt từ 1 đến N. Có $N-1$ cây cầu, mỗi cây cầu nối trực tiếp 2 hòn đảo tạo thành một quần thể đảo mà giữa hai hòn đảo bất kì luôn tồn tại đường đi thông qua một cây cầu hoặc một dãy các cây cầu.

Tuấn là một người giao hàng của công ty VOI. Ban đầu Tuấn ở hòn đảo 1, được giao K nhiệm vụ giao hàng theo thứ tự từ 1 đến K và Tuấn phải xử lý các nhiệm vụ theo đúng thứ tự đó. Với nhiệm vụ thứ i ($1 \le i \le K$), Tuấn cần thực hiện giao hàng từ hòn đảo U_i đến hòn đảo V_i theo đường đi ghé qua ít hòn đảo nhất. Tuấn chỉ có thể thực hiện nhiệm vụ giao hàng thứ i nếu vị trí của Tuấn đang ở U_i và sau khi hoàn thành giao hàng thì vị trí của Tuấn sẽ là V_i. Lưu ý đối với mỗi nhiệm vụ, Tuấn có thể thực hiện hoặc từ chối giao hàng. Đối với mỗi hòn đảo, công ty đặt một giá trị phần thưởng mà người giao hàng có thể nhận được cho mỗi lần ghé qua. Với mỗi nhiệm vụ hoàn thành, Tuấn sẽ nhận được phần thưởng là giá trị lớn nhất trong số các giá trị của các hòn đảo nằm trên đường đi giao hàng, tính cả hòn đảo xuất phát và hòn đảo kết thúc.

Yêu cầu: Hãy tính tổng giá trị phần thưởng lớn nhất mà Tuấn có thể nhận được sau K nhiệm vụ.

Dữ liệu

Vào từ file văn bản SHIP.INP:

- Dòng đầu chứa một số nguyên N là số lượng hòn đảo ($1 \le N \le 2 \times 10^5$).

- Dòng thứ hai chứa N số nguyên $W_1, W_2, \ldots, W_N$ là giá trị phần thưởng tại các hòn đảo ($0 \leq W_1, W_2, \ldots, W_N \leq 10^9$).
- Mỗi dòng trong số $N-1$ dòng tiếp theo chứa 2 số nguyên dương thể hiện cây cầu kết nối giữa 2 hòn đảo.
- Dòng tiếp theo chứa một số nguyên K là số lượng nhiệm vụ ($1 \leq K \leq 2 \times 10^5$).
- Dòng thứ i trong số K dòng tiếp theo chứa 2 số nguyên dương U_i và V_i thể hiện nhiệm vụ giao hàng thứ i ($1 \leq U_i, V_i \leq N; U_i \neq V_i$).

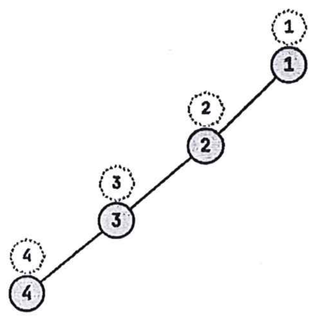
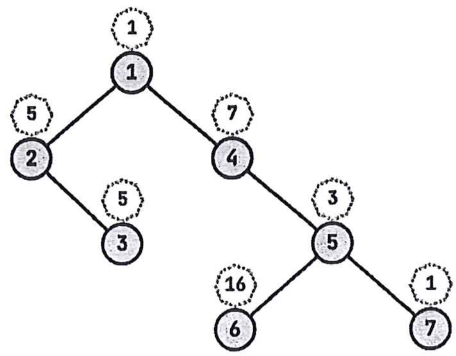
Các số trên cùng một dòng cách nhau bởi dấu cách.

Kết quả

Ghi ra file văn bản SHIP.OUT:

- Một số nguyên duy nhất thể hiện tổng giá trị phần thưởng lớn nhất mà Tuấn có thể nhận được sau K nhiệm vụ.

Ví dụ

SHIP.INP	SHIP.OUT	Giải thích
4 1 2 3 4 1 2 2 3 3 4 4 1 3 1 2 2 3 2 4	6	 - Tuấn thực hiện các nhiệm vụ 2, 4 và từ chối các nhiệm vụ 1, 3. - Tổng giá trị phần thưởng Tuấn nhận được là $2+4=6$.
7 1 5 5 7 3 16 1 1 4 1 2 2 3 4 5 5 6 5 7 7 1 3 3 2 1 4 4 2 2 5 5 7 5 6	37	 - Tuấn thực hiện các nhiệm vụ 3, 4, 5, 7 và từ chối các nhiệm vụ còn lại. - Tổng giá trị phần thưởng Tuấn nhận được là $7+7+7+16=37$.

Chấm điểm

- **Subtask 1** (20% số điểm): $N, K \leq 100$ và hòn đảo i có cây cầu nối với hòn đảo $i+1$ $(\forall i : 1 \leq i \leq N-1)$.
- **Subtask 2** (20% số điểm): $N \leq 10\,000$; $K \leq 100$ và hòn đảo i có cây cầu nối với hòn đảo $i+1$ $(\forall i : 1 \leq i \leq N-1)$.
- **Subtask 3** (20% số điểm): Hòn đảo i có cây cầu nối với hòn đảo $i+1$ $(\forall i : 1 \leq i \leq N-1)$.
- **Subtask 4** (20% số điểm): $N \leq 10\,000$ và $K \leq 100$.
- **Subtask 5** (20% số điểm): Không có ràng buộc nào thêm.

Bài 2. Lập trình Robot AI (7,0 điểm)

Trong cuộc thi trí tuệ nhân tạo ROAI2025, Ban tổ chức yêu cầu các đội chơi lập trình Robot AI trên sàn đấu trong môi trường thực tế ảo tham chiếu trên hệ trục tọa độ Oxy. Mỗi đội chơi cần đặt 2 con Robot AI tại 2 điểm phân biệt trên đường thẳng $y = 2$ với tọa độ x là số nguyên trong phạm vi từ 0 tới N. Mỗi Robot AI có một mắt thần có nhiệm vụ quan sát đường thẳng $y = 0$. Sàn đấu có hai tấm chắn biên là các đoạn thẳng AB và CD với tọa độ $A(0, 2)$, $B(0, 1)$, $C(N, 1)$, $D(N, 2)$. Ban tổ chức đặt thêm K tấm chắn khác là những đoạn thẳng nằm trên đoạn BC. Tấm chắn thứ i trong số K tấm chắn bắt đầu từ điểm $(L_i, 1)$ và kết thúc ở điểm $(R_i, 1)$. Robot AI tại điểm U trên đường thẳng $y = 2$ chỉ quan sát được những điểm V trên đường thẳng $y = 0$ nếu như đoạn thẳng UV không giao với bất kì đoạn thẳng là tấm chắn nào trong số $K + 2$ tấm chắn trên. Một đoạn trên đường thẳng $y = 0$ gọi là quan sát được nếu như mọi điểm trên đoạn thẳng đó (ngoại trừ 2 đầu mút) đều quan sát được. Nhiệm vụ của mỗi đội chơi là cần đặt được 2 con Robot AI sao cho tổng độ dài các đoạn quan sát được trên đường thẳng $y = 0$ bởi ít nhất một Robot AI là lớn nhất có thể.

Yêu cầu: Đội tuyển của Tuấn lần đầu tiên tham gia thi tài, hãy giúp đội của Tuấn tìm ra cách đặt tối ưu cho 2 con Robot AI.

Dữ liệu

Vào từ file văn bản ROAI.INP:

- Dòng đầu chứa hai số nguyên N và K $(1 \leq N \leq 10^9; 1 \leq K \leq 2\,000)$.
- Dòng thứ i trong số K dòng tiếp theo chứa hai số nguyên L_i và R_i $(0 \leq L_i < R_i \leq N)$.

Dữ liệu bảo đảm $L_1 < R_1 < L_2 < R_2 < \ldots < L_K < R_K$. Các số trên cùng một dòng cách nhau bởi dấu cách.

Kết quả

Ghi ra file văn bản ROAI.OUT:

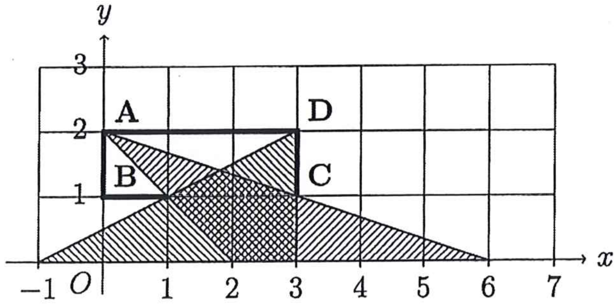
- Một số nguyên duy nhất là tổng độ dài các đoạn quan sát được trên đường thẳng $y = 0$ trong phương án tối ưu tìm được.

Ví dụ

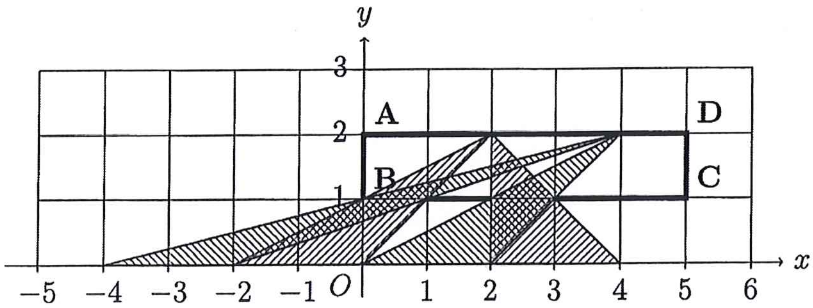
ROAI.INP	ROAI.OUT
3 1 0 1	7
5 2 1 2 3 5	8

Giải thích

- Trong ví dụ 1, một phương án đặt vị trí tối ưu cho 2 Robot AI là $x = 0$ và $x = 3$. Robot AI thứ nhất quan sát được đoạn $[2, 6]$; Robot AI thứ hai quan sát được đoạn $[-1, 3]$.



- Trong ví dụ 2, một phương án đặt vị trí tối ưu cho 2 Robot AI là $x = 2$ và $x = 4$. Robot AI thứ nhất quan sát được đoạn $[-2, 0]$ và đoạn $[2, 4]$; Robot AI thứ hai quan sát được đoạn $[-4, -2]$ và đoạn $[0, 2]$.



Chấm điểm

- Subtask 1 (16% số điểm): $N \le 200$.
- Subtask 2 (16% số điểm): $N \le 2\,000; K \le 20$.
- Subtask 3 (20% số điểm): $N \le 20\,000$.
- Subtask 4 (20% số điểm): $K \le 500$.
- Subtask 5 (28% số điểm): Không có ràng buộc nào thêm.

Bài 3. Số nguyên dương (6,0 điểm)

Thầy giáo Lê rất yêu thích các bài toán số học và dãy số. Thầy định nghĩa trọng số của một số nguyên dương X (ký hiệu là $W(X)$) như sau:

- Đầu tiên, thầy Lê biểu diễn X thành dãy các chữ số của nó ở hệ cơ số 10, thu được dãy $X_1, X_2, \ldots, X_N$ với N là số chữ số của X;
- Nếu dãy chữ số của X có chứa hai phần tử đứng cạnh nhau có tổng chia hết cho 5, nghĩa là $\exists i \in \{1, 2, \ldots, N-1\}$ sao cho $(X_i + X_{i+1}) \vdots 5$, thì $W(X) = 0$;
- Ngược lại, $W(X)$ chính là số lượng dãy con gồm các phần tử liên tiếp trong dãy chữ số của X mà có kết quả `xor` các phần tử bằng 0. Cụ thể, $W(X)$ là số lượng cặp i, j thoả mãn $1 \leq i \leq j \leq N$ và $X_i \oplus X_{i+1} \oplus \ldots \oplus X_j = 0$, với $\oplus$ là phép toán `xor` giữa hai số tự nhiên.

  Nhắc lại, với hai số tự nhiên a và b, phép toán $a \oplus b$ là phép toán `xor` từng bit tương ứng của a và b trong biểu diễn hệ nhị phân của chúng.

Ví dụ:

- $W(1234) = 0$ vì chứa hai chữ số 2 và 3 cạnh nhau có tổng chia hết cho 5;
- $W(122103) = 5$ vì không có hai chữ số nào cạnh nhau có tổng chia hết cho 5, đồng thời dãy chữ số của nó có 5 dãy con gồm các phần tử liên tiếp là $(2, 2)$, $(1, 2, 2, 1)$, $(1, 2, 2, 1, 0)$, (0) và $(2, 1, 0, 3)$ có `xor` các phần tử bằng 0;
- $W(3456) = 0$ vì dãy chữ số của nó không có dãy con nào có `xor` các phần tử bằng 0.

Yêu cầu: Cho số nguyên dương k và các số nguyên dương L, H. Hãy tính tổng luỹ thừa k của trọng số của tất cả các số nguyên dương nằm trong phạm vi $[L, H]$, nghĩa là tính $\sum_{X=L}^{H} (W(X))^k$.

Dữ liệu

Vào từ file văn bản PNUM.INP:

- Dòng đầu chứa một số nguyên dương k là số luỹ thừa trong tổng cần tính ($1 \leq k \leq 2$).
- Dòng thứ hai chứa một số nguyên dương T là số lượng trường hợp test ($1 \leq T \leq 50\,000$).
- Mỗi dòng trong số T dòng tiếp theo chứa hai số nguyên dương L và H mô tả một trường hợp test ($1 \leq L \leq H \leq 10^{1000}$).

Dữ liệu bảo đảm tổng số chữ số của tất cả các số L và H trong một file dữ liệu vào không vượt quá 10^5. Các số trên cùng một dòng cách nhau bởi dấu cách.

Kết quả

Ghi ra file văn bản PNUM.OUT:

- Gồm T dòng, mỗi dòng chứa một số nguyên là phần dư của kết quả tìm được trong phép chia cho 1 000 000 007 tương ứng với trường hợp test trong dữ liệu đầu vào.

Ví dụ

PNUM.INP	PNUM.OUT
1 3 1 10 100 105 111239 111245	1 5 12
2 3 1 10 100 105 111239 111245	1 7 30

Giải thích

Trong ví dụ thứ nhất:

- Ở trường hợp test thứ nhất: $W(1) = W(2) = W(3) = W(4) = W(5) = W(6) = W(7) = W(8) = W(9) = 0$, $W(10) = 1$, nên kết quả là $0^1+0^1+0^1+0^1+0^1+0^1+0^1+0^1+0^1+1^1 = 1$.
- Ở trường hợp test thứ hai: $W(100) = 0$, $W(101) = 2$, $W(102) = 1$, $W(103) = 1$, $W(104) = 1$, $W(105) = 0$, nên kết quả là $0^1 + 2^1 + 1^1 + 1^1 + 1^1 + 0^1 = 5$.
- Ở trường hợp test thứ ba: $W(111239) = 0$, $W(111240) = 3$, $W(111241) = 0$, $W(111242) = 2$, $W(111243) = 2$, $W(111244) = 3$, $W(111245) = 2$, nên kết quả là $0^1 + 3^1 + 0^1 + 2^1 + 2^1 + 3^1 + 2^1 = 12$.

Trong ví dụ thứ hai:

- Ở trường hợp test thứ nhất: Kết quả là $0^2+0^2+0^2+0^2+0^2+0^2+0^2+0^2+0^2+1^2 = 1$.
- Ở trường hợp test thứ hai: Kết quả là $0^2 + 2^2 + 1^2 + 1^2 + 1^2 + 0^2 = 7$.
- Ở trường hợp test thứ ba: Kết quả là $0^2 + 3^2 + 0^2 + 2^2 + 2^2 + 3^2 + 2^2 = 30$.

Chấm điểm

- Subtask 1 (12% số điểm): $T \leq 10$ và $H \leq 10^6$.
- Subtask 2 (12% số điểm): $T \leq 10$; $H \leq 10^{18}$ và $k = 1$.
- Subtask 3 (12% số điểm): $H \leq 10^{18}$ và $k = 1$.
- Subtask 4 (12% số điểm): $k = 1$.
- Subtask 5 (20% số điểm): $T \leq 10; H \leq 10^9$ và $k = 2$.
- Subtask 6 (16% số điểm): L là luỹ thừa của 10; $H = 10 \times L - 1$ và $k = 2$.
- Subtask 7 (16% số điểm): Không có ràng buộc nào thêm.

------------ HẾT ------------

* *Thí sinh* **KHÔNG** *được sử dụng tài liệu;*

* *Giám thị* **KHÔNG** *giải thích gì thêm.*

BỘ GIÁO DỤC VÀ ĐÀO TẠO

ĐỀ THI CHÍNH THỨC

KỲ THI CHỌN HỌC SINH GIỎI QUỐC GIA
TRUNG HỌC PHỔ THÔNG
NĂM HỌC 2024 - 2025

Môn: TIN HỌC
Thời gian: 180 phút (*không kể thời gian giao đề*)
Ngày thi thứ hai: 26/12/2024
Đề thi gồm 06 trang, 03 bài

TỔNG QUAN ĐỀ THI

	Tiêu đề	File chương trình	File dữ liệu	File kết quả
Bài 4	Vòng tròn	CYCLE.*	CYCLE.INP	CYCLE.OUT
Bài 5	Hai dãy số	TSEQ.*	TSEQ.INP	TSEQ.OUT
Bài 6	Mã hoá dãy số	ENCODE.*	ENCODE.INP	ENCODE.OUT

- Dấu * được thay thế bởi PAS hoặc CPP hoặc PY tương ứng với ngôn ngữ lập trình `Pascal` hoặc `C++` hoặc `Python`.
- Mỗi bài bao gồm nhiều subtask, mỗi subtask bao gồm nhiều test đơn, điểm của thí sinh được tính theo từng test đơn.

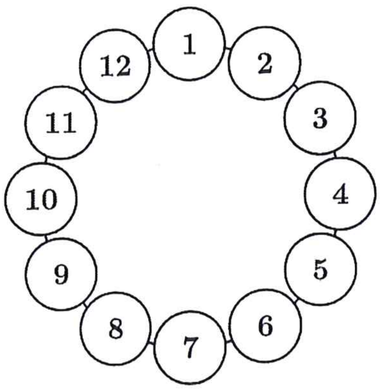
Hãy lập trình giải các bài toán sau:

Bài 4. Vòng tròn (7,0 điểm)

Một trường THPT chuyên đang tổ chức trò chơi tập thể cho M học sinh lớp chuyên Toán đánh số từ 1 đến M và L học sinh lớp chuyên Văn đánh số từ 1 đến L. Trên sân, có N vị trí cách đều nhau đã được đánh dấu sẵn, các vị trí được đánh số từ 1 đến N tạo thành một vòng tròn:

- Vị trí thứ 1 kề với vị trí thứ 2 và thứ N;
- Vị trí thứ i $(2 \le i \le N-1)$ kề với vị trí thứ $i-1$ và thứ $i+1$;
- Vị trí thứ N kề với vị trí thứ $N-1$ và thứ 1.

Hình vẽ dưới đây minh hoạ vòng tròn trên sân có 12 vị trí:



Ban đầu, $M + L$ học sinh đứng ở các vị trí sao cho không có 2 học sinh nào đứng cùng một vị trí. Học sinh thứ u của lớp chuyên Toán đứng ở vị trí P_u $(1 \leq u \leq M)$, học sinh thứ v của lớp chuyên Văn đứng ở vị trí Q_v $(1 \leq v \leq L)$. Mỗi bước, học sinh có thể di chuyển sang một trong 2 vị trí kề với vị trí đang đứng. Lưu ý, trong quá trình di chuyển, một vị trí có thể có nhiều hơn một học sinh. Các thầy cô muốn xác định vị trí tập trung của từng bạn sao cho:

- Vị trí tập trung của M học sinh lớp chuyên Toán là M vị trí liên tiếp ở trên vòng tròn;
- Vị trí tập trung của L học sinh lớp chuyên Văn là L vị trí liên tiếp trên vòng tròn;
- Không có 2 học sinh nào đứng cùng một vị trí.

Yêu cầu: Hãy giúp các thầy cô xác định vị trí tập trung của các học sinh sao cho tổng số bước phải di chuyển của tất cả các học sinh là nhỏ nhất.

Dữ liệu

Vào từ file văn bản CYCLE.INP:

Dòng đầu chứa một số nguyên T là số lượng trường hợp test $(1 \leq T \leq 10^5)$. Tiếp theo là T nhóm dòng, mỗi nhóm dòng mô tả một trường hợp test như sau:

- Dòng thứ nhất chứa 3 số nguyên dương N, M và L $(2 \leq M + L \leq N \leq 10^5)$.
- Dòng thứ hai chứa M số nguyên dương $P_1, P_2, \ldots, P_M$ $(1 \leq P_1, P_2, \ldots, P_M \leq N)$.
- Dòng thứ ba chứa L số nguyên dương $Q_1, Q_2, \ldots, Q_L$ $(1 \leq Q_1, Q_2, \ldots, Q_L \leq N)$.

Dữ liệu bảo đảm tổng các số N trong một file dữ liệu vào không vượt quá 2×10^5. Các số trên cùng một dòng cách nhau bởi dấu cách.

Kết quả

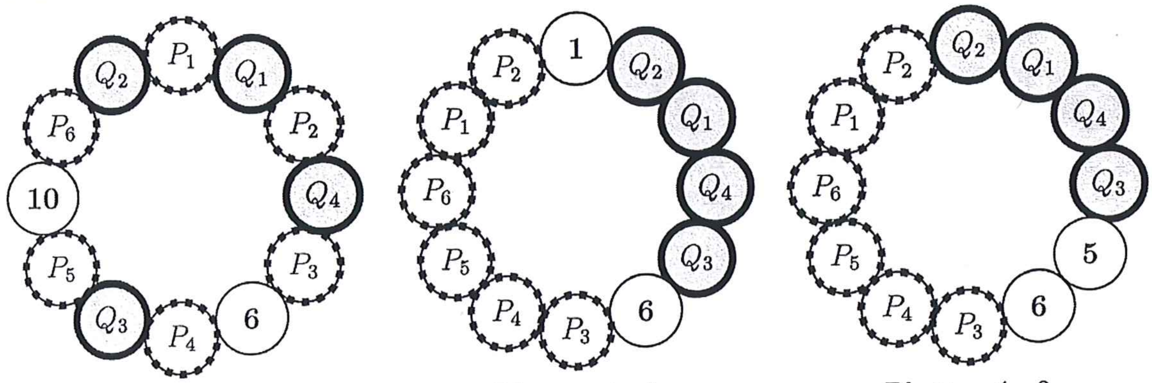
Ghi ra file văn bản CYCLE.OUT:

- Gồm T dòng, mỗi dòng chứa một số nguyên duy nhất là tổng số bước di chuyển nhỏ nhất của tất cả các học sinh tương ứng với trường hợp test trong dữ liệu vào.

Ví dụ

CYCLE.INP	CYCLE.OUT
1 12 6 4 1 3 5 7 9 11 2 12 8 4	15

Giải thích



Vị trí ban đầu — Phương án 1 — Phương án 2

- Trong phương án 1, số bước di chuyển của các học sinh chuyên Văn lần lượt là: 1,2,3,0; số bước di chuyển của các học sinh chuyên Toán lần lượt là: 2,3,2,1,0,1.
- Trong phương án 2, số bước di chuyển của các học sinh chuyên Văn lần lượt là: 0,1,4,1; số bước di chuyển của các học sinh chuyên Toán lần lượt là: 2,3,2,1,0,1.
- Cả 2 phương án đều tối ưu và đều có tổng số bước là 15.

Chấm điểm

- **Subtask 1** (10% số điểm): $N \le 10; T \le 10$.
- **Subtask 2** (15% số điểm): $N \le 500$; $1 \le P_1, P_2, \ldots, P_M < \frac{N}{2}$; $1 \le Q_1, Q_2, \ldots, Q_L < \frac{N}{2}$; $T \le 10$.
- **Subtask 3** (15% số điểm): $N \le 5000$; $1 \le P_1, P_2, \ldots, P_M < \frac{N}{2}$; $1 \le Q_1, Q_2 \ldots, Q_L < \frac{N}{2}$; $T \le 10$.
- **Subtask 4** (20% số điểm): $N \le 5000$; $T \le 10$.
- **Subtask 5** (20% số điểm): $1 \le P_1, P_2, \ldots, P_M < \frac{N}{2}$; $1 \le Q_1, Q_2, \ldots, Q_L < \frac{N}{2}$; $T \le 10$.
- **Subtask 6** (20% số điểm): Không có ràng buộc nào thêm.

Bài 5. Hai dãy số (7,0 điểm)

Với niềm đam mê về các bài toán trên dãy số của mình, hôm nay thầy giáo Lê nghiên cứu bài toán trên hai dãy số A và B. Dãy A gồm các số tự nhiên sắp xếp tăng dần từ 1 tới 10^9. Dãy B chỉ gồm duy nhất một số 0. Thầy Lê lần lượt đưa ra Q yêu cầu, mỗi yêu cầu thuộc một trong 3 loại sau:

- *Loại 1*: Cho hai số X và K, yêu cầu cắt K số đầu tiên của dãy A, ghép vào ngay sau số có giá trị X của dãy B;
- *Loại 2*: Cho hai số Y và H, yêu cầu xóa đi H số ngay sau số có giá trị Y của dãy B;
- *Loại 3*: Cho hai số L và R, yêu cầu tính tổng của các số từ vị trí L đến vị trí R của dãy B. Vị trí các phần tử trong dãy B được đánh số từ 1.

Yêu cầu: Hãy thực hiện các yêu cầu đó theo đúng thứ tự mà thầy Lê đưa ra.

Dữ liệu

Vào từ file văn bản TSEQ.INP:

- Dòng đầu chứa một số nguyên Q là số lượng yêu cầu thầy Lê cần thực hiện ($1 \le Q \le 5 \times 10^5$).
- Mỗi dòng trong số Q dòng tiếp theo chứa ba số nguyên mô tả một trong ba loại yêu cầu theo định dạng sau:

* 1 X K mô tả yêu cầu loại 1 ($0 \le X \le 10^9; 1 \le K \le 10^9$). Dữ liệu bảo đảm tổng K của tất cả các yêu cầu 1 không quá 10^9 và giá trị X hiện đang có trong dãy B;
* 2 Y H mô tả yêu cầu loại 2 ($0 \le Y \le 10^9; 1 \le H \le 10^9$). Dữ liệu bảo đảm có ít nhất H số sau giá trị Y và giá trị Y hiện đang có trong dãy B;
* 3 L R mô tả yêu cầu loại 3 ($1 \le L \le R \le |B|$, với $|B|$ là số lượng phần tử hiện tại của dãy B).

Các số trên cùng một dòng cách nhau bởi dấu cách.

Kết quả

Ghi ra file văn bản TSEQ.OUT:

- Với mỗi yêu cầu loại 3, ghi ra trên một dòng một số nguyên là đáp án tính được.

Ví dụ

TSEQ.INP	TSEQ.OUT
6 1 0 3 1 1 2 3 3 5 2 4 2 1 4 4 3 2 7	11 35

Giải thích

Yêu cầu	*Dãy B*	*Giải thích*
Ban đầu	[0]	
1	[0, 1, 2, 3]	
2	[0, 1, 4, 5, 2, 3]	
3	[0, 1, 4, 5, 2, 3]	Các số từ vị trí 3 đến 5 là [4, 5, 2] có tổng là 11
4	[0, 1, 4, 3]	
5	[0, 1, 4, 6, 7, 8, 9, 3]	
6	[0, 1, 4, 6, 7, 8, 9, 3]	Các số từ vị trí 2 đến 7 là [1, 4, 6, 7, 8, 9] có tổng là 35

Chấm điểm

- **Subtask 1** (15% số điểm): $Q, K \le 500$.
- **Subtask 2** (15% số điểm): $Q \le 5\,000$.
- **Subtask 3** (20% số điểm): Không có yêu cầu loại 2; Các yêu cầu loại 1 xuất hiện trước tất cả các yêu cầu loại 3.
- **Subtask 4** (25% số điểm): Không có yêu cầu loại 2.
- **Subtask 5** (25% số điểm): Không có ràng buộc nào thêm.

Bài 6. Mã hoá dãy số (6,0 điểm)

Ngày nay, lĩnh vực an ninh mạng đang dần trở thành một lĩnh vực thiết yếu trong việc bảo mật an toàn thông tin trên toàn thế giới. Tuấn là một kỹ sư an ninh mạng và rất đam mê khám phá các bài toán về mã hoá bảo mật. Cậu đang quan tâm đến bài toán mã hoá một dãy số $S = (S_1, S_2, \ldots, S_N)$ với các phần tử là các số nguyên dương không quá K. Cậu thử nghiệm việc mã hoá bằng cách viết ra tất cả các dãy hậu tố của dãy S. Cụ thể, với mỗi $i \in \{1, 2, \ldots, N\}$, Tuấn viết ra dãy $A_i = (S_i, S_{i+1}, \ldots, S_N)$. Sau đó, cậu sắp xếp các dãy này tăng dần theo thứ tự từ điển. Cuối cùng, cậu ghi lại độ dài của các dãy theo thứ tự đã sắp xếp, thu được dãy $L = (L_1, L_2, \ldots, L_N)$ và gọi đây là dãy đặc trưng của S.

Ví dụ, với dãy $S = (2, 1, 4, 3, 1, 3, 4)$ thì Tuấn sẽ viết ra 7 dãy: $A_1 = (2, 1, 4, 3, 1, 3, 4)$; $A_2 = (1, 4, 3, 1, 3, 4)$; $A_3 = (4, 3, 1, 3, 4)$; $A_4 = (3, 1, 3, 4)$; $A_5 = (1, 3, 4)$; $A_6 = (3, 4)$; $A_7 = (4)$. Sau khi sắp xếp tăng dần theo thứ tự từ điển, Tuấn thu được: $(A_5, A_2, A_1, A_4, A_6, A_7, A_3)$. Cuối cùng, Tuấn ghi lại độ dài các dãy theo thứ tự đó và thu được dãy đặc trưng của S là $L = (3, 6, 7, 4, 2, 1, 5)$.

Nhắc lại, dãy $(X_1, X_2, \ldots X_u)$ gọi là đi trước dãy $(Y_1, Y_2, \ldots, Y_v)$ theo thứ tự từ điển nếu một trong hai điều kiện sau thoả mãn:

- Tồn tại $i \in \{1, 2, \ldots, \min(u, v)\}$ sao cho $X_1 = Y_1, X_2 = Y_2, \ldots, X_{i-1} = Y_{i-1}$ và $X_i < Y_i$;
- $X_1 = Y_1, X_2 = Y_2, \ldots, X_u = Y_u$ và $u < v$.

Bây giờ Tuấn muốn biết khả năng giải mã ngược lại ra dãy S nếu chỉ biết số K, dãy L và phương pháp xây dựng dãy L. Tuấn nhận ra rằng có thể tồn tại nhiều dãy S có các phần tử là các số nguyên dương không quá K mà sau khi thực hiện phương pháp mã hoá như trên có thể cho ra cùng một dãy đặc trưng L.

Tiếp tục quá trình thử nghiệm, Tuấn thực hiện Q phép biến đổi liên tiếp trên dãy L. Với mỗi phép biến đổi, cậu sẽ chọn hai phần tử của dãy L hiện tại và đổi giá trị cho nhau, sau đó đếm số dãy S thoả mãn dãy đặc trưng L vừa được biến đổi. Sau mỗi phép biến đổi, dãy L được cập nhật.

Yêu cầu: Cho biết N, K, dãy L và các phép biến đổi của Tuấn. Trước lần biến đổi đầu tiên và sau mỗi lần biến đổi của Tuấn trên dãy L, hãy tính số lượng dãy S có N phần tử, các phần tử là các số nguyên dương không quá K, sao cho dãy đặc trưng của S là dãy L.

Dữ liệu

Vào từ file văn bản ENCODE.INP:

- Dòng đầu tiên chứa ba số nguyên N, K, Q $(1 \le N, K \le 5 \times 10^5; 0 \le Q \le 5 \times 10^5)$.
- Dòng thứ hai chứa N số nguyên dương $L_1, L_2, \ldots, L_N$. Dữ liệu bảo đảm dãy $L = (L_1, L_2, \ldots, L_N)$ là một hoán vị của N số tự nhiên $1, 2, \ldots, N$.
- Mỗi dòng trong số Q dòng tiếp theo chứa hai số nguyên dương khác nhau i, j $(1 \le i, j \le N)$ mô tả một phép biến đổi của Tuấn với ý nghĩa là đổi giá trị của L_i và L_j cho nhau.

Các số trên cùng một dòng cách nhau bởi dấu cách.

Kết quả

Ghi ra file văn bản ENCODE.OUT:

- Gồm $Q+1$ dòng, mỗi dòng là phần dư trong phép chia cho 1 000 000 007 của số lượng dãy S tính được cho dãy L ban đầu và cho dãy L sau mỗi phép biến đổi theo thứ tự đầu vào.

Ví dụ

ENCODE.INP	ENCODE.OUT	Giải thích
3 3 2 3 1 2 2 3 1 2	4 4 1	- Các dãy S có dãy đặc trưng $L=(3,1,2)$ là: $(1,2,2)$; $(1,3,3)$; $(2,3,3)$; $(1,3,2)$. - Các dãy S có dãy đặc trưng $L=(3,2,1)$ là: $(1,1,2)$; $(1,1,3)$; $(2,2,3)$; $(1,2,3)$. - Các dãy S có dãy đặc trưng $L=(2,3,1)$ là: $(2,1,3)$.

Chấm điểm

- **Subtask 1** (10% số điểm): $N, K, Q \leq 8$.
- **Subtask 2** (10% số điểm): $N, Q \leq 8$.
- **Subtask 3** (24% số điểm): $Q = 0$ và L là dãy tăng dần hoặc giảm dần.
- **Subtask 4** (28% số điểm): $N, K, Q \leq 1\,000$.
- **Subtask 5** (28% số điểm): Không có ràng buộc nào thêm.

------------ HẾT ------------

* *Thí sinh* **KHÔNG** *được sử dụng tài liệu;*

* *Giám thị* **KHÔNG** *giải thích gì thêm.*

BỘ GIÁO DỤC VÀ ĐÀO TẠO

ĐÁP ÁN

ĐỀ THI CHÍNH THỨC

KỲ THI CHỌN HỌC SINH GIỎI QUỐC GIA
TRUNG HỌC PHỔ THÔNG
NĂM HỌC 2024 - 2025

Môn: TIN HỌC
Ngày thi: 25-26/12/2024
(Đáp án gồm 08 trang)

TỔNG QUAN ĐỀ THI

	Tiêu đề	File chương trình	File dữ liệu	File kết quả
Bài 1	Người giao hàng	SHIP.*	SHIP.INP	SHIP.OUT
Bài 2	Lập trình Robot AI	ROAI.*	ROAI.INP	ROAI.OUT
Bài 3	Số nguyên dương	PNUM.*	PNUM.INP	PNUM.OUT
Bài 4	Vòng tròn	CYCLE.*	CYCLE.INP	CYCLE.OUT
Bài 5	Hai dãy số	TSEQ.*	TSEQ.INP	TSEQ.OUT
Bài 6	Mã hoá dãy số	ENCODE.*	ENCODE.INP	ENCODE.OUT

- Dấu * được thay thế bởi `PAS` hoặc `CPP` hoặc `PY` tương ứng với ngôn ngữ lập trình `Pascal` hoặc `C++` hoặc `Python`.
- Mỗi bài bao gồm nhiều subtask, mỗi subtask bao gồm nhiều test đơn, điểm của thí sinh được tính theo từng test đơn.

Bài 1. Người giao hàng (7,0 điểm)

- **Subtask 1** (20% số điểm): $N, K \leq 100$ và hòn đảo i có cây cầu nối với hòn đảo $i+1$ $(\forall i : 1 \leq i \leq N-1)$.

  Hòn đảo i có cây cầu nối với hòn đảo $i+1$, bài toán tìm giá trị lớn nhất trên đường đi giữa 2 hòn đảo u, v là bài toán tìm giá trị lớn nhất trên đoạn từ u đến v.

  Gọi $f[i, v]$ là giá trị phần thưởng có thể nhận được nhiều nhất sau nhiệm vụ i và Tuấn đứng ở hòn đảo v.

  Ở mỗi nhiệm vụ $i, 1 \leq i \leq K$, ta có $f[i, v_i] = \max(f[i-1, v_i], f[i-1, u_i] + maxPath(u_i, v_i))$, trong đó $maxPath(u_i, v_i)$ là giá trị lớn nhất trong đoạn từ u_i đến v_i được xử lý bằng cách tìm kiếm tuần tự.

  Độ phức tạp: $O(N^2 \times K)$.

- **Subtask 2** (20% số điểm): $N \leq 10\,000$; $K \leq 100$ và hòn đảo i có cây cầu nối với hòn đảo $i+1$ $(\forall i : 1 \leq i \leq N-1)$.

  Tương tự subtask 1. Thay tìm kiếm tuần tự trong $maxPath(u_i, v_i)$ bằng tìm kiếm dựa trên Cây phân đoạn hoặc Bảng thưa.

  Độ phức tạp: $O(N \times K \times \log N)$.

- **Subtask 3** (20% số điểm): Hòn đảo i có cây cầu nối với hòn đảo $i+1$ $(\forall i : 1 \le i \le N-1)$.

  Tương tự subtask 2. Gọi $f[v_i]$ là giá trị phần thưởng có thể nhận được nhiều nhất sau khi hoàn thành thực hiện nhiệm vụ và kết thúc ở hòn đảo v_i, cập nhật $f[v_i] = \max(f[v_i], f[u_i] + maxPath(u_i, v_i))$.

  Độ phức tạp: $O(N \times \log N + K)$.

- **Subtask 4** (20% số điểm): $N \le 10\,000$ và $K \le 100$.

  Tương tự subtask 3. Thay tìm kiếm tuần tự trong $maxPath(u_i, v_i)$ bằng BFS để tìm đường đi và giá trị lớn nhất giữa 2 hòn đảo u_i và v_i trên cây.

  Độ phức tạp: $O(N \times K)$.

- **Subtask 5** (20% số điểm): Không có ràng buộc nào thêm.

  Tương tự subtask 4. Thay BFS bằng kỹ thuật LCA và Bảng thưa để tính nhanh giá trị max trên đường đi giữa 2 hòn đảo u và v.

  Độ phức tạp: $O((N + K) \times \log N)$.

Bài 2. Lập trình Robot AI (7,0 điểm)

- **Subtask 1** (16% số điểm): $N \le 200$.

  Nhận thấy số lượng phương án của bài toán là $\frac{N\times(N+1)}{2}$. Với mỗi phương án, ta duyệt qua tất cả các đoạn đơn vị từ $-N$ tới $2 \times N$ để đánh dấu đoạn đơn vị được quan sát.

  Độ phức tạp tính toán là $O(N^3)$.

- **Subtask 2** (16% số điểm): $N \le 2\,000; K \le 20$.

  Cải tiến từ Subtask 1, thay vì duyệt tất cả các đoạn đơn vị, ta tìm ra danh sách đoạn của mỗi Robot. Do danh sách đã được sắp xếp tăng dần, ta có thể gộp các đoạn và tính tổng độ dài trong tối đa $2 \times K$ phép tính.

  Độ phức tạp tính toán chung là $O(N^2 \times K)$.

- **Subtask 3** (20% số điểm): $N \le 20\,000$.

  Nhận thấy, ta có thể cố định một Robot ở vị trí 0 mà không làm thay đổi lời giải tối ưu do tính chất tịnh tiến của bài toán. Tính chất tịnh tiến của bài toán là như sau: Phương án đặt hai Robot ở vị trí x_1 và x_2 sẽ tương đương với phương án đặt hai Robot ở vị trí $x_1 + \Delta$ và $x_2 + \Delta$.

  Do đó, số lượng phương án cần duyệt là $O(N)$.

  Độ phức tạp tính toán chung là $O(N \times K)$.

- **Subtask 4** (20% số điểm): $K \le 500$.

  Nếu phương án tối ưu không phải là phương án biên (Robot thứ nhất ở vị trí 0, Robot thứ hai ở vị trí N). Ta nhận thấy luôn tồn tại phương án tối ưu sao cho có ít nhất một đoạn quan sát được của Robot thứ nhất phải có mép (hai đầu mút L_i, R_i) trùng với một đầu mút (L_j hoặc R_j) của đoạn quan sát được của Robot thứ hai. Như vậy, bước đầu tiên, ra tìm ra tất cả khoảng cách thoả mãn điều kiện trên. Số lượng các khoảng cách thỏa mãn là $O(K^2)$. Với mỗi khoảng cách cố định, ta làm như subtask 2 để tìm ra độ dài quan sát được trong $O(K)$.

  Độ phức tạp tính toán là $O(K^3)$.

- **Subtask 5** (28% số điểm): Không có ràng buộc nào thêm.

  Ta xét bài toán tương đương sau:

  Cho hai danh sách, danh sách thứ nhất là danh sách các vùng không quan sát được của Robot thứ nhất (Robot thứ nhất đặt ở vị trí 0); danh sách thứ hai là danh sách các vùng quan sát được của Robot thứ hai. Do danh sách thứ hai là các đoạn không giao nhau nên ta có thể xử lý từng đoạn của danh sách thứ hai. Trước hết với mỗi đoạn $(U + \Delta, V + \Delta)$, ta đánh dấu được giá trị Δ_1^i là vị trí bắt đầu giao và Δ_2^i là vị trí kết thúc giao với đoạn $[L_i, R_i]$ của danh sách thứ nhất. Ngoài ra, ta cũng cần xét Δ_3^i và Δ_4^i vị trí bắt đầu và kết thúc của việc hai đoạn lồng vào nhau. Ta cần sắp xếp lại các Δ và đưa vào mảng đánh dấu theo thứ tự xuất hiện.

  Dùng kỹ thuật mảng cộng dồn và mảng hiệu, ta có thể tịnh tiến qua tất cả Δ. Mỗi lần tịnh tiến mất chi phí là $O(1)$.

  Độ phức tạp tính toán là $O(K^2 \log K)$.

Bài 3. Số nguyên dương (6,0 điểm)

- **Subtask 1:** $T \leq 10$ và $H \leq 10^6$.

  Tính trọng số của từng số từ 1 đến 10^6, sau đó tính tổng hoặc tổng bình phương trọng số các đoạn bằng cách sử dụng mảng cộng dồn.

  Độ phức tạp: $O(H \times \log^2(H) + T)$.

- **Subtask 2:** $T \leq 10$; $H \leq 10^{18}$ và $k = 1$.

  Đổi thứ tự tính: Xét lần lượt các cặp (i, j) mà $1 \leq i \leq j \leq 19$; sau đó đếm số lượng số x trong phạm vi $[L, H]$ thoả mãn $x_i \oplus x_{i+1} \oplus \ldots \oplus x_j = 0$; và không chứa hai chữ số nào cạnh nhau có tổng chia hết cho 5. Bài toán đếm này có thể giải quyết bằng quy hoạch động chữ số, có thể cài đặt bằng đệ quy và sử dụng mảng nhớ.

  Độ phức tạp: $O(T \times \log^3(H))$, với hằng số thuật toán là $16 \times 5 \times 10$.

- **Subtask 3:** $H \leq 10^{18}$ và $k = 1$.

  Tương tự subtask 2, nhưng xử lý bài toán đếm chung cho tất cả các trường hợp test; sau đó tìm thứ tự từ điển của H và $L - 1$ trong dãy nghiệm đếm được.

  Độ phức tạp phần tiền xử lý: $O(\log^3(H))$ với hằng số thuật toán là $16 \times 5 \times 10$.

  Độ phức tạp phần trả lời các trường hợp test: $O(N \times \log^2(H))$, với hằng số thuật toán là 10 và N là tổng số chữ số của tất cả các số L và H trong các trường hợp test.

- **Subtask 4:** $k = 1$.

  Thay vì xét từng cặp (i, j); có thể xét cấu hình nghiệm có dạng bộ hai số (x, y) với $x = x_1, x_2, \ldots, x_n$ là một dãy thập phân và $y = y_1, y_2, \ldots, y_n$ là một dãy nhị phân có các bit 1 kề nhau. Các cấu hình cần thoả mãn: x không chứa hai chữ số cạnh nhau có tổng chia hết cho 5; các bit 1 của y đánh dấu các chữ số tương ứng trên x sao cho `xor` của chúng bằng 0. Có thể đếm số cấu hình như vậy bằng cách đệ quy xây dựng từng vị trí của cả hai số (x, y) cùng một lúc; sau đó sử dụng mảng nhớ để khử trùng lặp các bài toán. Để kiểm soát phạm vi của x cho từng trường hợp test, có thể sử dụng kỹ thuật tìm thứ tự từ điển của nghiệm như sau: Để đếm số cấu hình bé hơn H, ta có thể xét i là vị trí đầu tiên mà $x_i < H_i$, và xét hết các khả năng của y có thể có tính đến i. Mặc dù số khả năng của y lớn, nhưng giá trị `xor` của các chữ số tương ứng là bé hơn 16; do đó có thể xét các giá trị `xor` này.

Độ phức tạp phần tiền xử lý: $O(\log(H))$ với hằng số thuật toán là $16 \times 5 \times 20$.

Độ phức tạp phần trả lời các trường hợp test: $O(N \times \log^2(H))$, với hằng số thuật toán là 20 và N là tổng số chữ số của tất cả các số L và H trong các trường hợp test.

- **Subtask 5:** $T \leq 10$; $H \leq 10^9$ và $k = 2$.

  Bình phương trọng số của một số $x = x_1, x_2, \ldots, x_n$ chính là số lượng bộ bốn số (i, j, k, t) sao cho $1 \leq i \leq j \leq n$; $1 \leq k \leq t \leq n$ và $x_i \oplus x_{i+1} \oplus \ldots \oplus x_j = x_k \oplus x_{k+1} \oplus \ldots \oplus x_t = 0$.

  Ý tưởng đổi thứ tự tính, tương tự như subtask 2: Xét tất cả các bộ bốn số (i, j, k, t) và đưa về đếm số lượng số x trong phạm vi $[L, H]$ mà $x_i \oplus x_{i+1} \oplus \ldots \oplus x_j = x_k \oplus x_{k+1} \oplus \ldots \oplus x_t = 0$; và không chứa hai chữ số nào cạnh nhau có tổng chia hết cho 5. Ta có thể giải quyết bằng quy hoạch động chữ số, cài đặt bằng đệ quy và sử dụng mảng nhớ.

  Độ phức tạp: $O(T \times \log^5(H))$, với hằng số thuật toán là $16 \times 16 \times 5 \times 10$.

- **Subtask 6:** L là luỹ thừa của 10 và $H = 10 \times L - 1$ và $k = 2$.

  Với ràng buộc này, các số x đều có số chữ số bằng nhau và có thể bỏ qua ràng buộc về phạm vi của x. Sử dụng cách đếm ở subtask 2; nhưng các số i, j, k, t có thể xây dựng bằng đệ quy thay vì vòng lặp (tương tự như cách xây dựng số x). Cấu hình cần đếm chính là bộ ba (x, y, z) trong đó $x = x_1, x_2, \ldots, x_n$ là một dãy các chữ số thập phân; $y = y_1, y_2, \ldots, y_n$ là một dãy nhị phân có các bit 1 nằm kề nhau; $z = z_1, z_2, \ldots, z_n$ là một dãy nhị phân có các bit 1 nằm kề nhau; sao cho các chữ số của x lấy tại các vị trí bit 1 của y có `xor` bằng 0 và các chữ số của x lấy tại các vị trí bit 1 của z có `xor` bằng 0. Có thể đếm số cấu hình như vậy bằng cách đệ quy xây dựng từng vị trí của cả ba số (x, y, z) cùng một lúc; sau đó sử dụng mảng nhớ để khử trùng lặp các bài toán. Để xử lý cho nhiều trường hợp thử nghiệm cùng một lúc, có thể cho phép số x có chữ số 0 đứng đầu.

  Độ phức tạp phần tiền xử lý: $O(\log(H))$ với hằng số thuật toán là $16 \times 16 \times 5 \times 40$.

  Độ phức tạp phần trả lời các trường hợp test: $O(N)$, với N là tổng số chữ số của tất cả các số L và H trong các trường hợp test.

- **Subtask 7:** Không có ràng buộc nào thêm.

  Với $k = 2$, sử dụng cách đếm tương tự như subtask 6 nhưng xử lý thêm ràng buộc về phạm vi của x. Để đếm số cấu hình bé hơn H, ta có thể xét i là vị trí đầu tiên mà $x_i < H_i$, và xét hết các khả năng các số y, z có thể có tính đến i. Mặc dù số khả năng của y và z lớn, nhưng giá trị `xor` của các chữ số tương ứng là bé hơn 16; do đó có thể xét các khả năng của các giá trị `xor` này, qua 16^2 trường hợp.

  Độ phức tạp phần tiền xử lý: $O(\log(H))$ với hằng số thuật toán là $16 \times 16 \times 5 \times 40$.

  Độ phức tạp phần trả lời các trường hợp test: $O(N)$, với hằng số thuật toán là $16 \times 16 \times 40$ và N là tổng số chữ số của tất cả các số L và H trong các trường hợp test.

Bài 4. Vòng tròn (7,0 điểm)

Không mất tính tổng quát, giả thiết chiều tăng dần của chỉ số vị trí là chiều kim đồng hồ.

- **Subtask 1** (10% số điểm): $N \leq 10$; $T \leq 10$.

  Đối với mỗi học sinh, duyệt mọi trạng thái kết thúc có thể.

  Sau khi duyệt toàn bộ $L + M$ học sinh, với mỗi trạng thái kết thúc, kiểm tra điều kiện mất $O(N)$.

  Tổng độ phức tạp là $O(N! \times N)$.

- **Subtask 2** (15% số điểm): $N \leq 500$; $1 \leq P_1, \ldots, P_M < \frac{N}{2}$; $1 \leq Q_1, \ldots, Q_L < \frac{N}{2}$; $T \leq 10$.

  Ta ký hiệu:

  * *bl*$[u]$ là chi phí tối thiểu để sắp xếp L bạn lớp Văn liên tiếp nhau theo chiều kim đồng hồ bắt đầu từ u.
  * *bm*$[u]$ là chi phí tối thiểu để sắp xếp M bạn lớp Toán liên tiếp nhau theo chiều kim đồng hồ bắt đầu từ u.

  Ta thực hiện việc tính mảng *bm*, *bl* độc lập với nhau. Không mất tính tổng quát, ta xét việc tính *bm* cho dãy $P_1, \ldots, P_M$ đã sắp xếp tăng dần.

  Do P_i đều bé hơn $\frac{N}{2}$ nên di chuyển từ P_i tới v bất kỳ, ta không bao giờ phải sử dụng bước đi từ N tới 1. Tuy nhiên, ta vẫn có thể bước đi từ 1 đến N.

  Gọi k là số bạn phải sử dụng bước đi $1 \to N$. Ta có nhận xét sau:

  - Với $u < N/2$, $k = 0$.
  - Với $u \geq N/2$, $k \leq \min(N - u, M)$.

  Do vậy với mỗi $u < \frac{N}{2}$, ta cho lần lượt $P_1, \ldots, P_M$ vào các vị trí bắt đầu từ u theo chiều kim đồng hồ và tính khoảng cách tương ứng. Độ phức tạp tính toán: $O(M)$.

  Còn với $u \geq \frac{N}{2}$, ta duyệt và tìm giá trị k cho tổng khoảng cách bé nhất. Độ phức tạp tính toán: $O(M^2)$.

  Do vậy, chi phí tính toán ra *bm* và *bl* là $O(N^3)$.

  Đáp án của bài toán có thể tính trong $O(N^2)$ bằng cách duyệt mọi vị trí bắt đầu đoạn liên tiếp của hai lớp.

  Tổng độ phức tạp là $O(N^3)$.

- **Subtask 3** (15% số điểm): $N \leq 5\,000$; $1 \leq P_1, \ldots, P_M < \frac{N}{2}$; $1 \leq Q_1, \ldots, Q_L < \frac{N}{2}$; $T \leq 10$.

  Ta sử dụng:

  * *bm_head*$[k]$ là chi phí để k bạn có vị trí bé nhất dồn về vị trí từ 1 tới k.
  * *bm_tail*$[k]$ là chi phí để k bạn có vị trí lớn nhất dồn về vị trí từ N tới $N - k + 1$.

  Sử dụng hai mảng tính sẵn này, chi phí di chuyển với u, k cho trước có thể tính trong $O(1)$. Độ phức tạp tính toán cho việc tìm ra *bm*$[u]$ là $O(M)$.

  Tổng độ phức tạp tính toán là $O(N^2)$.

- **Subtask 4** (20% số điểm): $N \leq 5\,000$; $T \leq 10$.

  Ta thực hiện việc tính mảng bm, bl với ý nghĩa như ở subtask 3.

  Tuy nhiên, ta cần thực hiện như sau:

  – Chia tập học sinh lớp chuyên toán thành 2 tập, di chuyển theo chiều kim đồng hồ và di chuyển ngược chiều kim đồng hồ.
  – Việc chia tập có N^2 khả năng. Gọi p, q là điểm đầu và điểm cuối của tập di chuyển ngược chiều kim đồng hồ, ta có thể tính:
    * tổng khoảng cách để dồn toàn bộ học sinh từ p đến q về phía p.
    * tổng khoảng cách để dồn toàn bộ học sinh từ $q+1$ đến $p-1$ về phía $p-1$.
    * tính được cách giá trị $bm[x]$ với x nằm trong khoảng $p-1$ và p.

  Độ phức tạp thuật toán $O(N^2)$.

- **Subtask 5** (20% số điểm): $1 \leq P_1, \ldots, P_M < \frac{N}{2}$; $1 \leq Q_1, \ldots, Q_L < \frac{N}{2}$; $T \leq 10$.

  Tư tưởng tương tự như subtask 3 nhưng chúng ta cần tăng tốc tính toán mảng bm và bl.

  Gọi k_u^{opt} là giá trị giúp tổng khoảng cách là bé nhất. Với $\frac{N}{2} \leq u < N$, ta có $0 \leq k_{u+1}^{opt} \leq k_u^{opt}$. Do đó, ta cập nhật dần các phương án tối ưu cho trường hợp $u \geq \frac{N}{2}$ trong $O(1)$. Độ phức tạp cho việc tính bm là $O(N)$.

  Áp dụng STL deque, ta có thể tính đáp án bài toán từ bm và bl trong $O(N)$.

  Tổng độ phức tạp là $O(N)$.

- **Subtask 6** (20% số điểm): Không có ràng buộc nào thêm.

  Tư tưởng tương tự như subtask 4 nhưng chúng ta cần tăng tốc tính toán mảng bm và bl.

  Để làm được việc này, ta phân tích quá trình tính toán $bm[x]$. Khi tính giá trị $bm[x]$ ta cần quan tâm 3 tập sau:

  – Có 1 số học sinh di chuyển thuận chiều kim đồng hồ (tập A);
  – Có 1 số học sinh giữ nguyên vị trí (tập B);
  – Có 1 số học sinh di chuyển ngược chiều kim đồng hồ (tập C).

  Ta có nhận xét:

  – Khi tính giá trị $bm[x+1]$, tổng khoảng cách sẽ tăng lên 1 lượng $|A| + |B| - |C|$.
  – Sau khi tính $bm[x+1]$, ta phải cập nhật lại các tập A, B, C.
  – Ban đầu, ta có thể khởi tạo ngẫu nhiên nhát cắt p, q.
  – Lưu ý rằng mỗi học sinh chỉ đổi từ C sang B, B sang A đúng 1 lần.

  Áp dụng STL deque, ta có thể tính mảng bm trong $O(N)$.

  Tổng độ phức tạp là $O(N)$.

Bài 5. Hai dãy số (7,0 điểm)

- **Subtask 1** (15% số điểm): $Q, K \le 500$.

  Làm đúng như yêu cầu đề bài. Độ dài tối đa của dãy B là $Q \times K$. Độ phức tạp của mỗi yêu cầu là $O(Q \times K)$.

  Độ phức tạp chung là $O(Q^2 \times K)$.

- **Subtask 2** (15% số điểm): $Q \le 5\,000$.

  Thay vì lưu từng phần tử, ta lưu các đoạn vào cấu dữ liệu danh sách. Độ dài danh sách là $O(Q)$. Thế nên, việc quản lý danh sách sẽ mất độ phức tạp là $O(Q)$.

  Độ phức tạp chung là $O(Q^2)$.

- **Subtask 3** (20% số điểm): Không có yêu cầu loại 2 và các yêu cầu loại 1 xuất hiện trước tất cả các yêu cầu loại 3.

  Dùng con trỏ để truy được vị trí các đoạn (ℓ, r) trên danh sách. Mỗi thao tác chèn, ta tìm đối tượng (ℓ, r) trên danh sách. Sau đó, ta xóa đi và chèn lại các đoạn thay thế.

  Sau khi có được danh sách trên, ta dùng mảng cộng dồn để tính kết quả mỗi yêu cầu loại 3.

  Độ phức tạp chung là $O(Q \log Q)$.

- **Subtask 4** (25% số điểm): Không có yêu cầu loại 2.

  Xử lý offline để xây dựng được danh sách các đoạn (ℓ, r). Sử dụng cây Fenwick để cập nhật độ dài khi được thêm và yêu cầu loại 3.

  Độ phức tạp chung là $O(Q \log Q)$.

- **Subtask 5** (25% số điểm): Không có ràng buộc nào thêm.

  Ta phải xử lý thêm thao tác xóa bằng kỹ thuật Nâng nhị phân trên cây Fenwick (Binary Lifting on Fenwick tree).

  Độ phức tạp vẫn là $O(Q \log Q)$.

Bài 6. Mã hoá dãy số (6,0 điểm)

- **Subtask 1** (10% số điểm): $N, K, Q \le 8$.

  Xây dựng tất cả k^n dãy s khác nhau có thể có và kiểm tra từng dãy.

  Độ phức tạp: $O(Q \times K^N \times N \log N)$.

- **Subtask 2** (10% số điểm): $N, Q \le 8$.

  Giả sử dãy s chỉ chứa các giá trị $\{1, 2, \ldots, t\}$ và mỗi giá trị đều xuất hiện ít nhất một lần. Khi đó có thể xây dựng tất cả các dãy s có thể có; số lượng dãy s như vậy là không quá n^n. Kiểm tra từng dãy s, nếu s có dãy đặc trưng là L thì kết quả sẽ tăng thêm $\binom{k}{t}$ (chính là số cách để chọn ra t số trong k số được phép).

  Độ phức tạp: $O(Q \times N^N \times N \log N)$.

- **Subtask 3** (24% số điểm): $Q = 0$ và L là dãy tăng dần hoặc giảm dần.

  Trong subtask này s là một dãy đơn điệu, việc đếm số cấu hình s sẽ tương tự như một bài toán chia kẹo Euler.

  Độ phức tạp: $O(N)$.

- **Subtask 4** (28% số điểm): $N, K, Q \leq 1\,000$.

  Với mỗi $i \in \{1, 2, \ldots, n-1\}$; đặt $x = l[i]$ và $y = l[i+1]$. Do $A_x < A_y$ nên $S_{n-x+1} \leq S_{n-y+1}$. Việc dấu bằng có thể xảy ra hay không, phụ thuộc vào việc so sánh A_{x-1} với A_{y-1}. Từ cấu hình của dãy L, ta có thể biết kết quả so sánh giữa A_{x-1} với A_{y-1}, và suy ra ràng buộc ở đây là $S_{n-x+1} \leq S_{n-y+1}$ hay $S_{n-x+1} < S_{n-y+1}$. Gọi c là số vị trí i mà dấu bằng có thể xảy ra. Khi đó kết quả bài toán là $\sum_{i=0}^{c} \binom{c}{i} \times \binom{k}{n-i}$.

  Độ phức tạp: $O(N \times Q)$.

- **Subtask 5** (28% số điểm): Không có ràng buộc nào thêm.

  Tương tự như subtask 4, nhưng cần tối giản công thức. Kết quả là: $\binom{k+c}{n}$. Với mỗi truy vấn, số lượng vị trí i mà ràng buộc chuyển từ "$\leq$" sang "$<$" (hoặc ngược lại) là ít, do đó có thể cập nhật lại giá trị c một cách nhanh chóng.

  Độ phức tạp: $O(N + Q)$.

-------------- HẾT --------------