TAM GIÁC

Cho ma trận a gồm m hàng và n cột, các hàng đánh số từ trên xuống dưới, các cột đánh số từ trái sang phải. Ô ở hàng i cột j có giá trị nguyên $a_{i,j}$ ($|a_{i,j}| \leq 10^9$). Với mỗi ma trận vuông $k \times k$ định nghĩa *nửa dưới* của nó là tất cả các ô nằm trên đường chéo chính và dưới đường chéo chính trái trên – phải dưới.

Yêu cầu: Tìm hình vuông cỡ $k \times k$ sao cho tổng các số ở nửa dưới của nó đạt giá trị lớn nhất.

Dữ liệu: Vào từ file văn bản **TRIANGLE.INP**

- Dòng đầu ghi 3 số m, n, k ($m, n \leq 2000, k \leq \min(m, n)$)
- m dòng tiếp theo, mỗi dòng gồm n số là ma trận a.

Kết quả: Đưa ra file văn bản **TRIANGLE.OUT**

Ghi 1 số nguyên duy nhất là kết quả tìm được.

Ví dụ:

TRIANGLE.INP	TRIANGLE.OUT
3 4 2 1 2 **1 1** 2 1 **3 4** 1 2 1 2	8

Các giới hạn:

- Sub 1 (30%): $m, n \leq 50$
- Sub 2 (40%): $50 < m, n \leq 500$
- Sub 3 (30%): $500 < m, n \leq 2000$.

ĐÈN LỒNG

Trên trục đường phố chính của thành phố Thái Nguyên có n tòa nhà, được đánh số theo thứ tự từ 1 đến n, tòa nhà thứ i có chiều cao là một số nguyên dương h_i.

Để chuẩn bị cho Mùa du lịch Thái Nguyên năm nay, thành phố lập kế hoạch treo đèn lồng trang trí cho các tòa nhà. Hai tòa nhà liền kề i và $i+1$ mất chi phí $c \times |h_i - h_{i+1}|$ (c là hằng số). *Chi phí của trục đường* là tổng chi phí của các tòa nhà kề nhau, tức là $S = c \times \sum_{i=1}^{n-1} |h_i - h_{i+1}|$. Thành phố đẹp nhất khi các con đường đều đẹp nhất. Tuy nhiên, do nguồn kinh phí có hạn, lãnh đạo thành phố quyết định chọn giải pháp cho tu sửa nâng chiều cao một số ngôi nhà để tiết kiệm chi phí, cụ thể nếu tòa nhà i nâng chiều cao thêm x (đơn vị, $x > 0$) thì thành phố phải mất một khoản chi phí là x^2.

Yêu cầu: Cho biết n, c và các chiều cao h_i ($i = 1..n$), bạn hãy giúp thành phố tính chi phí S thấp nhất khi thực hiện theo kế hoạch nhé.

Dữ liệu: Vào từ file văn bản **MICOST.INP**

- Dòng đầu tiên chứa hai số nguyên n, c ($1 \le n \le 10^4, 1 \le c \le 10^6$);
- Mỗi dòng trong n dòng sau chứa một số nguyên h_i ($1 \le h_i \le 1000$).

Hai số liên tiếp trên cùng dòng được ghi cách nhau bởi dấu cách.

Kết quả: Đưa ra file văn bản **MICOST.OUT** một số nguyên là chi phí thấp nhất mà thành phố phải trả.

Ví dụ:

MICOST.INP	MICOST.OUT
5 2 2 3 5 1 4	15

Giải thích: Nâng tòa nhà 1 thêm 1, nâng tòa nhà 4 thêm 2. Khi đó chiều cao các tòa nhà lần lượt là: $3, 3, 5, 3, 4$.

Tổng chi phí là: $2 \times (0 + 2 + 2 + 1) + 1^2 + 2^2 = 15$.

Ràng buộc:

- Sub 1 (30%): $n \le 10$; $h_i \le 3$
- Sub 2 (40%): $n \le 1000$; $h_i \le 100$
- Sub 3 (30%): $n \le 10000$; $h_i \le 1000$.

TRUY VẤN TRÊN ĐỒ THỊ

Cho n đội chè đánh số từ 1 đến n và $n-1$ đường đi trực tiếp sao cho từ một đội chè luôn có đường đi tới một đội chè khác. Chi phí đi từ đội chè i đến đội chè j là một số nguyên dương c_{ij}.

Một đường đi đơn từ đội chè u đến đội chè v là dãy $u = x_1 x_2 \dots x_k = v$ trong đó $(x_i, x_{i+1}), i = 1..(k-1)$ là đường đi trực tiếp và với mọi $i, j: x_i \neq x_j$. Chi phí của đường đi trên là giá trị nhỏ nhất của các đường nối trực tiếp giữa hai đội chè kề nhau nằm trên đường đi đó.

Yêu cầu: Cho Q truy vấn, mỗi truy vấn được mô tả bởi hai số nguyên k, v với ý nghĩa: Đếm xem có bao nhiêu đội chè u mà đường đi đơn từ đội chè u đến đội chè v có chi phí không nhỏ hơn k.

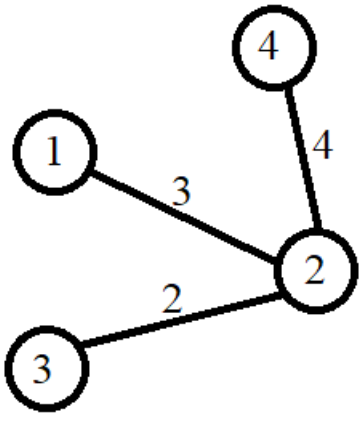
Dữ liệu: Vào từ file văn bản **QKGRAPH.INP**

- Dòng thứ nhất chứa hai số nguyên dương n, Q $(1 \leq n, Q \leq 10^5)$
- $n-1$ dòng sau mô tả các đường nối trực tiếp giữa các đội chè. Dòng thứ i chứa ba số nguyên dương p_i, q_i, c_i thể hiện có đường đi từ đội chè p_i đến đội chè q_i với chi phí c_i $(1 \leq p_i, q_i \leq n, 1 \leq c_i \leq 10^9)$.
- Q dòng tiếp theo, mỗi dòng ghi một truy vấn gồm hai số nguyên k, v $(1 \leq k \leq 10^9, 1 \leq v \leq n)$ thể hiện yêu cầu đếm xem có bao nhiêu đội chè mà chi phí đường đi đơn từ nó đến đội chè v không nhỏ hơn k.

Kết quả: Ghi ra file văn bản **QKGRAPH.OUT**

Gồm Q dòng, mỗi dòng ghi một số nguyên là kết quả của truy vấn tương ứng (theo thứ tự xuất hiện trong file dữ liệu vào).

Ví dụ:



QKGRAPH.INP	QKGRAPH.OUT
4 3 1 2 3 2 3 2 2 4 4 1 2 4 1 3 1	3 0 2

Các giới hạn:

- Sub 1 (30%): $n \leq 10^5, Q = 1$
- Sub 2 (30%): $n \leq 1000, Q \leq 10^3$
- Sub 3 (40%): $n, Q \leq 10^5$.

XORGAND

Một số M đươc gọi là số huyền bí cơ số X nếu (M ^ X) > (M & X). Trong đó ^ là phép XOR bit, & là phép AND bit trong C++.

Bạn được cho 1 mảng N số nguyên và Q truy vấn. Mỗi truy vấn được cho bởi bộ 3 số L, R và X. Yêu cầu tìm trong đoạn từ L đến R xem có bao nhiêu số huyền bí cơ số X.

Dữ liệu vào:

Dòng đầu chứa số nguyên N.

Dòng tiếp theo chứa N số nguyên là các phần tử của mảng.

Dòng thứ ba chứa số nguyên Q là số truy vấn.

Q dòng cuối cùng, mỗi dòng chứa 3 số nguyên L, R và X biểu thị truy vấn.

Dữ liệu ra:

Q dòng là kết quả cho từng truy vấn.

Ví dụ:

XORGAND.INP	XORGAND.OUT
5 1 2 3 4 5 2 1 5 4 2 5 2	3 2

Ghi chú:

Subtask 1: 50% số test có N và Q <= 2000.

Subtask 2: 50% số test còn lại có N và Q <= 200000.

SẮP XẾP

Nhận thấy quân địch đang trong tình trạng suy kiệt cả về số lượng lẫn chất lượng, Darth Vader đã ra lệnh cho đô đốc Sakerm lập tức chuẩn bị quân đội để tổng tấn công.

Hiện tại, đô đốc Sakerm đang gấp rút chuẩn bị quân đội. Quân đội của ông có N người lính, mỗi tên lính trong hàng ngũ đều có một chỉ số riêng biệt. Dãy chỉ số của N người lính là một hoán vị của dãy số nguyên từ 1 đến N. Để đoàn quân tinh nhuệ của ông được tối ưu sức mạnh, Sakerm phải chỉnh đốn lại hàng ngũ sao cho chỉ số của các người lính trong hàng tăng dần.

Trong quá khứ, ông thường dùng các thuật toán Bubble Sort để chỉnh đốn hàng ngũ. Tuy nhiên do đang gấp rút và thuật toán kia lại tốn quá nhiều thời gian nên buộc Sakerm phải tìm cách sắp xếp khác. May mắn cho ông, tất cả các lính tinh nhuệ của ông đều đã được rèn luyện bài bản, chỉ cần gọi tên thì ngay lập tức 1 chuỗi hành động sau sẽ được người lính vừa được gọi tên ấy thực hiện:

- Nếu người lính bên phải có chỉ số nhỏ hơn người lính được gọi thì 2 người lính này sẽ đổi chỗ cho nhau và quá trình này sẽ lặp đi lặp lại đến khi không còn người lính bên phải có chỉ số nhỏ hơn.
- Ngược lại, nếu người lính bên trái có chỉ số lớn hơn người lính được gọi thì 2 người lính này sẽ đổi chỗ cho nhau và quá trình này sẽ lặp đi lặp lại đến khi không còn người lính bên trái có chỉ số lớn hơn.

Mỗi người lính khi được gọi tên sẽ thực hiện chuỗi hành động trên chỉ tốn 1 giây.

Sakerm nhận thấy rằng không nhất thiết phải gọi tên toàn bộ N người lính mà chỉ cần gọi 1 số người trong đó thì toàn bộ hàng ngũ sẽ được sắp xếp. Sakerm không giỏi tính toán nên ông rất muốn biết xem số lượng tên lính tối thiểu ông cần gọi tên là bao nhiêu. Ngoài ra, có thể có nhiều tập thỏa mãn có cùng số lượng phần tử. Là 1 người ham học hỏi, Sakerm tò mò muốn biết tập bé thứ K trong thứ tự từ điển là tập nào. Bạn hãy giúp Sakerm sắp xếp quân ngũ nhằm chuẩn bị tổng phản công nhé. (Dữ liệu đảm bảo luôn có nhiều hơn K tập thỏa mãn).

Input:

- Dòng đầu chứa 2 số nguyên $N \leq 10^5$ và $K \leq 10^{18}$
- Dòng thứ hai chứa N số nguyên là hoán vị của dãy số từ 1 đến N.

Output:

- Dòng đầu chứa số nguyên x là số lượng người lính cần gọi tên ít nhất.
- x dòng sau, mỗi dòng chứa 1 số nguyên thuộc tập bé thứ K, các số sắp xếp theo thứ tự từ điển cần tìm (liệt kê theo thứ tự tăng dần).

SAPXEP.INP	SAPXEP.OUT
5 1 3 5 1 4 2	3 1 2 4

Subtask

- 20% số test có $N \leq 6$ và $K = 1$.
- 30% số test tiếp theo có $N \leq 10^5$ và $K = 1$
- 50% số test còn lại không có điều kiện gì thêm

Giải thích:

- Sau khi gọi tên người lính có chỉ số 1 (tức là ở vị trí 3 trong dãy ban đầu), dãy trở thành **1 3 5 4 2**.
- Sau khi gọi tên người lính có chỉ số 2, dãy trở thành **1 2 3 5 4**.
- Sau khi gọi tên người lính có chỉ số 4, dãy trở thành **1 2 3 4 5**.

Như vậy, sau 3 lần gọi tên, quân đoàn đã được sắp xếp, và không có cách nào có số lần gọi bé hơn cũng như thứ tự từ điển bé hơn **1 2 4**.

TÊN TRỘM

Giới hạn thời gian: 2 giây, Bộ nhớ 512Mb

Tên trộm OBI thâm nhập vào trung tâm điều khiển của thị trấn A. Trung tâm điều khiển này có dạng 1 đồ thị N đỉnh vô hướng liên thông không có chu trình. Mỗi đỉnh tương ứng là 1 phòng và các cạnh là đường hầm nối 2 phòng đó. Những phòng chỉ có duy nhất 1 đường hầm nối với nó sẽ có 1 lối ra (vào) – được coi là nút lá trên đồ thị.

Biết có tên trộm đột nhập, giám đốc trung tâm điều khiển điều động các lính canh tại các lối thoát hiểm nhằm truy bắt tên trộm. Tên trộm cố gắng di chuyển ra một nút lá nào đó để thoát thân; các lính canh cố gắng di chuyển để chặn được tên trộm.

Bạn hãy tính giúp ông giám đốc xem với mỗi phòng từ i đến N, nếu tên trộm dịch chuyển ra phòng i thì cần tối thiểu bao nhiêu tên lính để chặn đứng tên trộm. Giả sử tên trộm và các lính canh đều di chuyển tối ưu: tên trộm di chuyển tối ưu để cực tiểu hóa số tên lính mình phải đối đầu. Còn lính canh luôn di chuyển về phía tên trộm. Mỗi 1 bước di chuyển sẽ đi qua được 1 cạnh. Các lính canh sẽ được đi vào qua các nút lá.

INPUT:

- Dòng đầu chứa số nguyên N ($N<=7*10^4$)
- N-1 dòng sau, mỗi dòng chứa 2 số nguyên u, v thể hiện có đường hầm nối giữa 2 phòng u và v (u, v <=n).

OUTPUT:

Ghi ra n dòng, dòng thứ i là số tên lính tối thiểu cần để chặn đứng tên trộm nếu hắn dịch chuyển ra phòng thứ i.

Ví dụ:

Tentrom.inp	Tentrom.out
6 1 2 1 3 2 4 3 5 5 6	2 2 2 1 2 1

Ghi chú:

Subtask 1: 20% số test có N<=6

Subtask 2: 20% số test có N<=2000

Subtask 3: 60% số test có $N<=7*10^4$