Câu 1. *(7,0 điểm)* **TRANH CỬ**

Ở đất nước TST, có n thành phố được đánh số từ 1 đến n. Các thành phố được kết nối với nhau bởi $n-1$ con đường. Người dân tại nước đó di chuyển giữa các thành phố thông qua những con đường này. Họ có thể đi lại theo cả hai chiều trên mỗi con đường và di chuyển giữa hai thành phố bất kỳ bằng cách đi qua một hoặc nhiều con đường.

Ông John là một ứng cử viên sáng giá cho chức vụ tổng thống. Để chiến thắng trong cuộc bầu cử, ông cần tiến hành một chiến dịch tranh cử quy mô lớn. Thư ký của ông đã lập ra m kế hoạch cho chiến dịch tranh cử. Trong kế hoạch thứ i $(1 \leq i \leq m)$, ông sẽ di chuyển từ thành phố A_i đến thành phố B_i thông qua số lượng con đường đi ít nhất có thể, đồng thời thực hiện một bài diễn thuyết công khai tại mỗi thành phố trên lộ trình đó (bao gồm cả thành phố A_i và B_i).

Theo tính toán của thư ký, nếu kế hoạch thứ i được thực hiện thành công, ông sẽ thu về được c_i phiếu bầu (kế hoạch thất bại nếu trên lộ trình diễn thuyết tồn tại một thành phố ông đã từng diễn thuyết tại kế hoạch khác). Ông có thể lựa chọn thực hiện nhiều kế hoạch tranh cử. Số phiếu bầu đạt được là tổng số phiếu bầu thu được sau khi thực hiện các kế hoạch tranh cử ông đã chọn.

Vì muốn trở thành tổng thống, ông quyết tâm giành được càng nhiều phiếu bầu càng tốt. Với tư cách là một lập trình viên xuất sắc tại nước TST, bạn được ông ấy chọn và yêu cầu viết một chương trình tính số phiếu bầu tối đa mà ông có thể nhận được trong cuộc bầu cử tổng thống thông qua m kế hoạch đề ra.

Dữ liệu vào: Đọc từ tệp tin văn bản **TRANHCU.INP** có cấu trúc như sau:

- Dòng đầu tiên ghi số nguyên n ($2 \le n \le 10^5$) là số lượng thành phố tại nước TST;
- $n-1$ dòng tiếp theo, mỗi dòng ghi 2 số nguyên X_i và Y_i ($1 \le X_i, Y_i \le n$, $X_i \ne Y_i, 1 \le i \le n-1$) cách nhau bởi dấu cách cho biết con đường thứ i nối trực tiếp giữa hai thành phố X_i và Y_i;
- Dòng tiếp theo ghi số nguyên m ($1 \le m \le 10^5$) là số lượng kế hoạch tranh cử được thư ký đề xuất;
- m dòng tiếp theo, mỗi dòng ghi 3 số nguyên lần lượt là A_i, B_i và c_i ($1 \le A_i, B_i \le n, A_i \ne B_i, 1 \le c_i \le 10^4, 1 \le i \le m$) cách nhau bởi dấu cách cho biết kế hoạch thứ i đi từ thành phố A_i đến B_i và thu được c_i phiếu bầu.

Dữ liệu ra: Ghi ra tệp tin văn bản **TRANHCU.OUT** một số nguyên là tổng số phiếu bầu tối đa mà ông John có thể nhận được.

Ví dụ:

TRANHCU.INP	TRANHCU.OUT
7 3 4 6 5 2 7 1 5 7 5 4 5 5 4 3 10 5 6 5 2 6 9 7 2 2 1 3 8	19

Ràng buộc:

- Có 25% số test tương ứng 25% số điểm với $m \le 15$;
- Có 20% số test tương ứng 20% số điểm với $X_i = i, Y_i = i+1$ ($1 \le i \le n-1$), $c_i = 1$ ($1 \le i \le m$);
- Có 15% số test tương ứng 15% số điểm với $X_i = i, Y_i = i+1$ ($1 \le i \le n-1$);
- Có 15% số test tương ứng 15% số điểm với $c_i = 1 (1 \le i \le m)$;
- Có 15% số test tương ứng 15% số điểm với $n, m \le 10^3$;
- Có 10% số test tương ứng 10% số điểm không có ràng buộc gì thêm.

Câu 2. ***(7,0 điểm)*** **TÔ MÀU**

Cho một cây n đỉnh được đánh số từ 1 đến n. Ban đầu mọi đỉnh đều chưa được tô màu. Bạn cần thực hiện q truy vấn, mỗi truy vấn thuộc một trong hai loại sau:

- Loại 1 "set x y" ($1 \le x, y \le n$): Tô đỉnh x bằng màu y. Dữ liệu đảm bảo ngay trước truy vấn này đỉnh x đang chưa được tô màu hoặc đỉnh x có thể đã từng được tô màu ở một truy vấn trước đó, nhưng màu đó đã bị một truy vấn loại 2 hủy đi.
- Loại 2 "rollback": Hủy truy vấn loại 1 gần nhất mà hiện tại vẫn còn hiệu lực (tức là đỉnh được tô màu trong truy vấn đó trở lại trạng thái chưa có màu).

Lưu ý:

- Truy vấn loại 2 hủy đi truy vấn tô màu cuối cùng còn hiệu lực, không phải quay lại trạng thái ngay trước truy vấn liền trước. Có thể có nhiều truy vấn loại 2 đứng liền nhau, khi đó chúng lần lượt hủy các truy vấn tô màu còn hiệu lực theo thứ tự từ mới nhất đến cũ nhất.

- Dữ liệu đảm bảo rằng tại mọi thời điểm, mỗi màu xuất hiện trên không quá hai đỉnh của cây.

Đỉnh v là một đỉnh tốt nếu mọi đường đi đơn xuất phát từ v không tồn tại hai đỉnh tô cùng một màu. Các đỉnh chưa được tô màu không mang màu nào nên không bao giờ gây ra trùng lặp.

Yêu cầu: Sau mỗi truy vấn loại 1, hãy đếm số đỉnh tốt của cây tại thời điểm đó.

Dữ liệu vào: Đọc từ tệp tin văn bản **TOMAU.INP** có cấu trúc như sau:

- Dòng đầu tiên ghi hai số nguyên dương n và q ($1 \le n, q \le 3.10^5$) lần lượt là số đỉnh của cây và số truy vấn;
- $n-1$ dòng tiếp theo, mỗi dòng ghi hai số nguyên u và v ($1 \le u, v \le n$) mô tả một cạnh của cây nối đỉnh u và đỉnh v;
- q dòng cuối cùng, mỗi dòng mô tả một truy vấn là một trong hai dạng "set x y" hoặc "rollback".

Dữ liệu ra: Ghi ra tệp tin văn bản **TOMAU.OUT**

Với mỗi truy vấn loại 1, in ra trên một dòng một số nguyên là số đỉnh tốt của cây sau khi thực hiện truy vấn đó.

Ví dụ 1:

TOMAU.INP	TOMAU.OUT
3 6 1 2 1 3 set 1 3 set 2 3 set 3 2 rollback rollback set 3 2	3 0 0 3

Ví dụ 2:

TOMAU.INP	TOMAU.OUT
5 8 1 2 2 3 1 4 2 5 set 1 1 set 5 1 rollback rollback set 1 1 set 3 1 set 5 2 set 4 2	5 2 5 2 2 1

Ràng buộc:

- *Có 15% số test tương ứng 15% số điểm với* $n \leq 2.10^2$*;*
- *Có 10% số test tương ứng 10% số điểm với* $n \leq 2.10^3$ *và không có truy vấn loại 2;*
- *Có 15% số test tương ứng 15% số điểm với* $n \leq 2.10^3$*;*
- *Có 20% số test tương ứng 20% số điểm với Cây có dạng đường thẳng;*
- *Có 15% số test tương ứng 15% số điểm không có truy vấn loại 2;*
- *Có 25% số test tương ứng 25% số điểm không có ràng buộc gì thêm.*

Câu 3. *(6,0 điểm)* **XẾP HẠNG**

Trên mặt phẳng tọa độ Oxy, cho n điểm được đánh số thứ tự từ 1 đến n. Điểm thứ i có tọa độ $(x[i], x[i]^2)$; biết các giá trị của $x[i]$ là số nguyên, đôi một khác nhau. Điểm thứ i hoạt động tại thời điểm t $(t \in Z)$ khi và chỉ khi $L[i] \leq t \leq R[i]$.

Bạn được cho q truy vấn, mỗi truy vấn gồm 4 số nguyên (t, dx, dy, k). Hệ số ưu tiên cho điểm thứ i xác định bằng công thức:

$$s[i] = dx.x[i] + dy.x[i]^2 \ (|s[i]| < 2^{63})$$

Ta tiến hành xếp hạng các điểm đang hoạt động tại thời điểm t theo hệ số ưu tiên giảm dần; nếu bằng hệ số ưu tiên, điểm nào có số thứ tự bé hơn thì được xếp trước. Hãy tìm số thứ tự của điểm được xếp ở hạng thứ k. Nếu số điểm đang hoạt động nhỏ hơn k thì in ra -1.

Dữ liệu vào: Đọc từ tệp tin văn bản **XEPHANG.INP** có cấu trúc như sau:

- Dòng đầu tiên ghi số n $(1 \leq n \leq 2.10^5)$;
- n dòng tiếp theo mỗi dòng ghi 3 số nguyên lần lượt là $x[i], L[i], R[i]$ $(|x[i]| \leq 10^6, 1 \leq L[i] \leq R[i] \leq 10^9)$;
- Dòng tiếp theo ghi số nguyên q $(1 \leq q \leq 2.10^5)$ là số truy vấn;

• q dòng tiếp theo mỗi dòng ghi lần lượt 4 số nguyên t, dx, dy, k ($1 \le t \le 10^9$; $|dx|, |dy| \le 10^6$; $(dx, dy) \ne (0,0)$; $1 \le k \le n$).

Dữ liệu ra: Ghi ra tệp tin văn bản **XEPHANG.OUT** gồm q dòng, dòng thứ i ghi một số nguyên là kết quả của truy vấn thứ i.

Ví dụ:

XEPHANG.INP	XEPHANG.OUT
4 -2 1 6 0 1 6 3 2 5 5 4 6 4 1 1 0 1 3 0 -1 2 5 1 1 1 6 0 -1 3	2 1 4 4

Giải thích ví dụ:

Với truy vấn thứ 3: $t = 5;\ dx = 1;\ dy = 1;\ k = 1$. Tại thời điểm t cả bốn điểm đang hoạt động. Khi đó ta tính hệ số ưu tiên của các điểm như sau:

$s[1] = 2; s[2] = 0; s[3] = 12; s[4] = 30$. Vậy điểm xếp hạng thứ k là 4.

Ràng buộc:

- *Có 10% số test tương ứng 10% số điểm với $n, q \le 20$;*
- *Có 15% số test tương ứng 15% số điểm với $n, q \le 3.10^3$;*
- *Có 15% số test tương ứng 15% số điểm với $n, q \le 12.10^3$;*
- *Có 20% số test tương ứng 20% số điểm với $n, q \le 5.10^4$;*
- *Có 40% số test tương ứng 40% số điểm với $n, q \le 2.10^5$.*

-----Hết-----

Họ và tên thí sinh:..... Số báo danh:.....

Giám thị 1:.....Ký tên ..... Giám thị 2:.....Ký tên .....