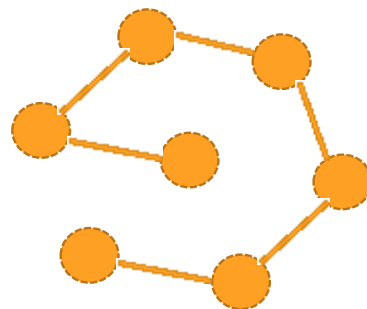


**04 BÀI TẬP VỀ CÂY ĐỒ THỊ (TREE) (T2)****1. Tiền bối bậc 2^t**

Cho cây T gồm n đỉnh được đánh số thứ tự $1, 2, \dots, n$. Gốc là đỉnh 1. Với mỗi đỉnh u thuộc cây T , sẽ có một đường đi duy nhất từ đỉnh 1 đến đỉnh u . Giả sử đường đi đó là:

$$1 = u_0 \rightarrow u_1 \rightarrow u_2 \rightarrow \dots \rightarrow u_m = u.$$

Khi đó ta nói:

$1 = u_0$ là tiền bối bậc m của u .

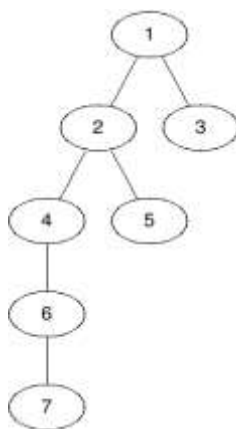
u_1 là tiền bối bậc $m - 1$ của u .

u_2 là tiền bối bậc $m - 2$ của u .

u_t là tiền bối bậc $m - t$ của u .

Như vậy đỉnh v được gọi là tiền bối bậc k của u ($k \leq m$) nếu đường đi từ v đến u có đúng k cạnh.

Ví dụ:



Với cây ở trên, ta có: 6 là tiền bối bậc 1 của 7; 4 là tiền bối bậc 2 của 7; 2 là tiền bối bậc 3 của 7; 1 là tiền bối bậc 4 của 7.

Yêu cầu: Cho đỉnh u và t , hãy tìm đỉnh tiền bối bậc 2^t của đỉnh u .

Dữ liệu cho trong file Ance2T.Inp gồm:

- Dòng đầu ghi số nguyên dương n là số đỉnh của cây.
- Tiếp theo gồm $n - 1$ dòng, mỗi dòng gồm hai số u, v mô tả một cạnh của cây.
- Dòng tiếp theo ghi số q là số các truy vấn.
- q dòng cuối, mỗi dòng ghi hai số nguyên t, u mô tả truy vấn tìm đỉnh tiền bối bậc 2^t của u ($0 \leq t \leq 20; 1 \leq u \leq n$).

Kết quả ghi ra file Ance2T.Out gồm q dòng, mỗi dòng là kết quả của một truy vấn tương ứng trong file dữ liệu. Nếu không có đỉnh tiền bối bậc 2^t của đỉnh u thì ghi kết quả là -1 .



Ví dụ:

Ance2T.Inp	Ance2T.Out	Hình minh họa
7 1 2 1 3 2 4 2 5 4 6 6 7 3 0 7 1 5 2 7	6 1 -1	

Giới hạn:

- Sub1: $n, q \leq 10^3$;
- Sub2: $n, q \leq 2 \cdot 10^5$;



2. Tiền bối bậc k

Cho cây T gồm n đỉnh được đánh số thứ tự $1, 2, \dots, n$. Gốc là đỉnh 1. Với mỗi đỉnh u thuộc cây T , sẽ có một đường đi duy nhất từ đỉnh 1 đến đỉnh u . Giả sử đường đi đó là:

$$1 = u_0 \rightarrow u_1 \rightarrow u_2 \rightarrow \dots \rightarrow u_m = u.$$

Khi đó ta nói:

$1 = u_0$ là tiền bối bậc m của u .

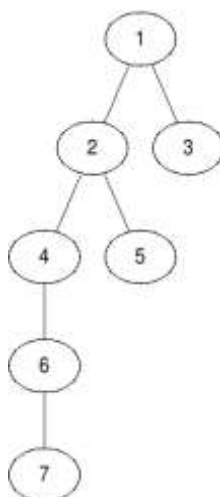
u_1 là tiền bối bậc $m - 1$ của u .

u_2 là tiền bối bậc $m - 2$ của u .

u_t là tiền bối bậc $m - t$ của u .

Như vậy đỉnh v được gọi là tiền bối bậc k của u ($k \leq m$) nếu đường đi từ v đến u có đúng k cạnh.

Ví dụ:





Với cây ở trên, ta có: 6 là tiền bối bậc 1 của 7; 4 là tiền bối bậc 2 của 7; 2 là tiền bối bậc 3 của 7; 1 là tiền bối bậc 4 của 7.

Yêu cầu: Cho đỉnh u và k , hãy tìm đỉnh tiền bối bậc k của đỉnh u .

Dữ liệu cho trong file AnceK.Inp gồm:

- Dòng đầu ghi số nguyên dương n là số đỉnh của cây.
- Tiếp theo gồm $n - 1$ dòng, mỗi dòng gồm hai số u, v mô tả một cạnh của cây.
- Dòng tiếp theo ghi số q là số các truy vấn.
- q dòng cuối, mỗi dòng ghi hai số nguyên k, u mô tả truy vấn tìm đỉnh tiền bối bậc k của u .

Kết quả ghi ra file AnceK.Out gồm q dòng, mỗi dòng là kết quả của một truy vấn tương ứng trong file dữ liệu. Nếu không có đỉnh tiền bối bậc k của đỉnh u thì ghi kết quả là -1 .

Ví dụ:

AnceK.Inp	AnceK.Out	Hình minh họa
7 1 2 1 3 2 4 2 5 4 6 6 7 4 3 7 1 5 3 5 0 3	2 2 -1 3	

Giới hạn:

- Sub1: $n, q \leq 10^3$;
- Sub2: $n, q \leq 2 \cdot 10^5$;



3☀. Tiền bối – Hậu duệ

Cho cây $T = (V, E)$ gồm n đỉnh, các đỉnh được đánh số từ 1 đến n . Đỉnh 1 là gốc của cây.

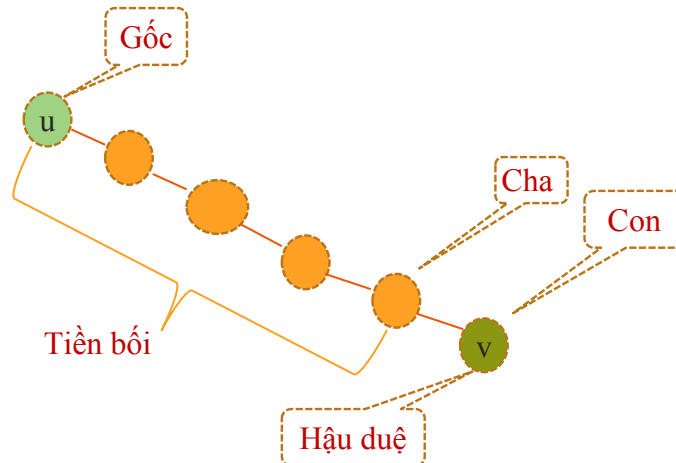
Mối quan hệ “Tiền bối – Hậu duệ” được định nghĩa như sau:

Với một đỉnh v thuộc T có gốc u . Khi đó có đường đi duy nhất từ đỉnh gốc u đến v .

Giả sử đường đi qua các đỉnh: $u = u_1 \rightarrow u_2 \rightarrow u_3 \rightarrow \dots \rightarrow u_{k-2} \rightarrow u_{k-1} \rightarrow u_k = v$.

☛ Ta gọi: Đỉnh u_{k-1} là **đỉnh cha** của đỉnh v ; v là **đỉnh con** của đỉnh u_{k-1} .

☛ Các đỉnh: u_1, u_2, \dots, u_{k-1} được gọi là các **đỉnh tiền bối** của v , v là **đỉnh hậu duệ** của các đỉnh u_1, u_2, \dots, u_{k-1} .



Yêu cầu: Với hai đỉnh u, v . Hãy kiểm tra xem u, v có mối quan hệ “*Tiền bối – Hậu duệ*” hay không? Tức là u có phải là tiền bối của v hay không? Hoặc v có phải là tiền bối của u hay không?

Dữ liệu cho trong file **TREEANCE.INP** như sau:

- Dòng đầu ghi hai số nguyên dương n và q tương ứng là số đỉnh và số truy vấn.
- $n - 1$ dòng tiếp theo, mỗi dòng gồm hai số u, v mô tả cạnh (u, v) của cây.
- q dòng cuối, mỗi dòng ghi hai số nguyên x, y , cần xét mối quan hệ “*Tiền bối – Hậu duệ*” của cặp đỉnh này.

Kết quả ghi ra file **TREEANCE.OUT** gồm q dòng, nếu x, y là “*Tiền bối – Hậu duệ*” của nhau thì ghi ra 1, ngược lại ghi ra 0.

Giới hạn:

- Sub1: $n \leq 1000, q \leq 100$;
- Sub2: $n \leq 10^5, q \leq 10^5$.

Ví dụ:

TREEANCE.INP	TREEANCE.OUT	Hình minh họa
5 3 1 2 1 5 2 3 2 4 1 4 3 2 4 5	1 1 0	

**4☀. Nút cha chung gần nhất (The Lowest Common Ancestor - LCA)**

Cho một cây có n đỉnh được đánh số từ 1 đến n . Cây có gốc là đỉnh 1. Có m truy vấn, mỗi truy vấn cần trả lời nút cha chung của cả hai đỉnh u và v đồng thời nút cha chung đó xa nút gốc nhất có thể (tức là gần u, v nhất có thể).

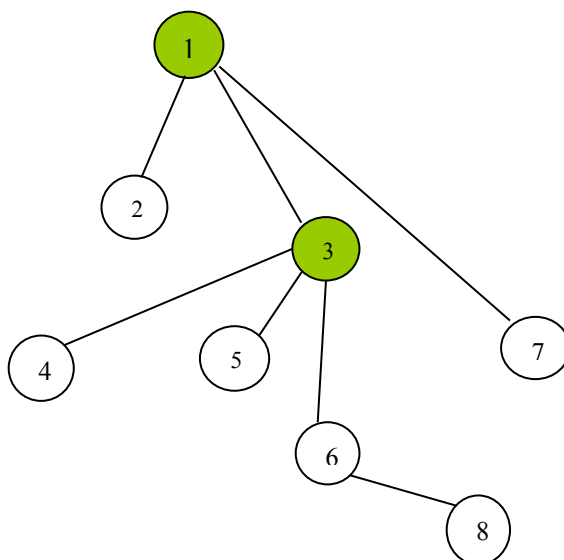
Dữ liệu cho trong file văn bản LCA.INP như sau:

- Dòng đầu tiên ghi hai số nguyên dương n, m ($3 \leq n, m \leq 10^5$) tương ứng là số đỉnh trên cây và số truy vấn.
- $n - 1$ dòng sau mỗi dòng ghi hai số nguyên dương u, v mô tả cạnh u, v trên cây.
- m dòng cuối cùng, mỗi dòng ghi hai số u, v mô tả truy vấn cần đưa ra nút cha chung gần nhất của đỉnh u và v .

Kết quả ghi ra file LCA.OUT gồm m dòng, dòng thứ i là kết quả truy vấn của truy vấn thứ i .

Ví dụ:

LCA.INP	LCA.OUT
8 3	1
1 2	3
1 3	1
1 7	
3 4	
3 5	
3 6	
6 8	
1 2	
4 8	
5 7	



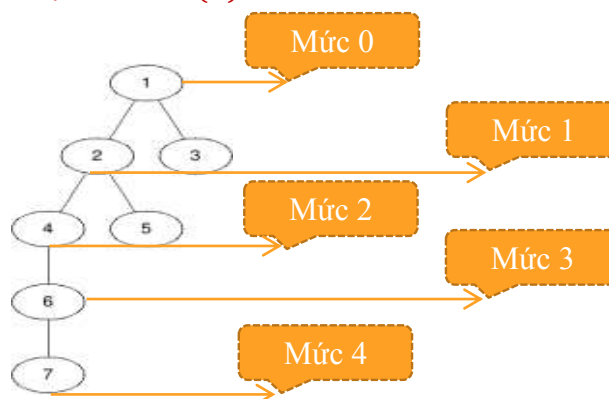
**5☀. Di chuyển trên cây**

Cho cây T gồm n đỉnh được đánh số thứ tự $1, 2, \dots, n$. Gốc là đỉnh 1. Với mỗi đỉnh u thuộc cây T , sẽ có một đường đi duy nhất từ đỉnh 1 đến đỉnh u . Giả sử đường đi đó là:

$$1 = u_0 \rightarrow u_1 \rightarrow u_2 \rightarrow \dots \rightarrow u_m = u.$$

Ta gọi mức của đỉnh u là m và ta kí hiệu là $Level(u) = m$.

Ví dụ:



Có hai con robot được đặt tại hai đỉnh u và v khác nhau của cây. Cứ trong một giây, robot có thể di chuyển sang một đỉnh kề cạnh, nhưng robot cũng có thể đứng yên mà không di chuyển.

Yêu cầu: Tính thời gian ít nhất để, robot thứ nhất di chuyển từ đỉnh u đến u_1 ; robot thứ hai di chuyển từ v đến v_1 sao cho u_1, v_1 là hai đỉnh có cùng mức. Chú ý là: u có thể bằng u_1 ; v có thể bằng v_1 và u_1, v_1 có thể bằng nhau.

Dữ liệu cho trong file ROTREE.INP gồm:

- Dòng đầu ghi số nguyên dương n là số đỉnh của cây.
- $n - 1$ dòng tiếp theo, mỗi dòng ghi 2 số nguyên e, f mô tả cạnh của cây.
- Dòng tiếp theo ghi số nguyên dương q là số trường hợp thực hiện di chuyển hai con robot.
- q dòng cuối mỗi dòng ghi hai số u, v ($u \neq v$) là vị trí đặt hai con robot.

Kết quả ghi trong file ROTREE.OUT gồm q dòng. Mỗi dòng ghi thời gian ít nhất để di chuyển hai robot đến hai đỉnh có cùng mức trong cây (ứng với cặp robot trong file dữ liệu).

Ví dụ:

ROTREE.INP	ROTREE.OUT	Hình minh họa
7	0	
1 2	1	
1 3		
2 4		
2 5		
4 6		
6 7		
2		
4 5		
2 6		



6☀. Xóa nút trên cây

Cho cây T gồm n đỉnh, các đỉnh được đánh số từ 1, 2, ..., n , đỉnh 1 là đỉnh gốc. Có q phép thực hiện xóa các đỉnh trên cây, mỗi đỉnh xóa một đỉnh lá của cây. Sau mỗi lần xóa, hãy tính độ dài đường đi dài nhất trên cây T (độ dài của đường đi là số cạnh trên đường đi đó).

Dữ liệu cho trong file DelTree.Inp gồm:

- Dòng đầu ghi hai số nguyên dương n và q ($1 \leq q < n \leq 10^5$).
- $n - 1$ dòng sau, mỗi dòng ghi hai số nguyên dương u và v mô tả một cạnh của cây.
- Dòng cuối ghi q số u_1, u_2, \dots, u_q . Dữ liệu đảm bảo, trước khi bắt đầu xóa đỉnh u_i , đỉnh u_i là một đỉnh lá.

Kết quả ghi ra file DelTree.Out gồm q dòng. Dòng thứ i ghi độ dài đường đi dài nhất trên cây khi xóa đỉnh u_i .

Ví dụ:

DelTree.Inp	DelTree.Out	Hình minh họa
8 4 1 2 1 3 2 4 2 5 4 6 5 7 3 8 6 7 8 3	5 4 3 2	

**7☀. Đường đi đơn điệu trên cây**

Cho cây gồm n đỉnh và $n - 1$ cạnh có trọng số khác nhau. Tìm một cặp đỉnh (u, v) sao cho đường đi từ u đến v : $u \rightarrow u_1 \rightarrow u_2 \rightarrow u_3 \rightarrow \dots \rightarrow u_k = v$ có a_1, a_2, \dots, a_k tạo thành dãy đơn điệu (tăng hoặc giảm) dài nhất (k lớn nhất); trong đó a_1 là trọng số cạnh (u, u_1) , a_2 là trọng số cạnh (u_1, u_2) , .., a_k là trọng số cạnh (u_{k-1}, u_k) .

- Dãy a_1, a_2, \dots, a_k được gọi là đơn điệu tăng nếu $a_1 < a_2 < a_3 < \dots < a_k$.
- Dãy a_1, a_2, \dots, a_k được gọi là đơn điệu giảm nếu $a_1 > a_2 > a_3 > \dots > a_k$.

Dữ liệu: Vào từ file văn bản PathIncTree.Inp gồm:

- Dòng đầu ghi số nguyên dương n ($n \leq 10^5$).
- $n - 1$ dòng tiếp theo, mỗi dòng ghi 3 số nguyên dương u, v, c mô tả cạnh (u, v) có trọng số c của cây ($1 \leq c \leq 10^6$). Chú ý là, trọng số của $n - 1$ cạnh khác nhau.

Kết quả: ghi ra file PathIncTree.Out là độ dài (số cạnh) của đường đi từ đỉnh u đến v tìm được.

Ví dụ:

PathIncTree.Inp	PathIncTree.Out	Hình minh họa
6 1 2 2 1 3 4 3 4 1 3 5 6 5 6 3	3	 Đường đi đỉnh: $2 \rightarrow 1 \rightarrow 3 \rightarrow 5$ có trọng số: 2, 4, 6 tạo thành dãy tăng dần.

**8☀. Cây máy tính**

Trung tâm máy tính của phòng nghiên cứu và phát triển AI của đất nước XYZ có n máy tính. Các máy tính được đánh số thứ tự $1, 2, \dots, n$ và có $n - 1$ cáp nối cho phép truyền tin hai chiều. Mỗi cáp nối được nối trực tiếp với hai máy tính sao cho hai máy bất kì đều có thể truyền tin cho nhau. Nghĩa là các máy tính được kết nối với nhau tạo thành mô hình cây (tree). Máy tính 1 được xem là máy trung tâm (và là gốc của cây). Các máy tính trong mạng có mối quan hệ máy cha, máy con như quan hệ cha – con của các nút trong cây. Máy thứ i có hiệu năng tính toán là T_i .

Có q phép kiểm tra và điều chỉnh các thông số về hiệu năng tính toán trên các máy tính.

Phép kiểm tra thứ nhất có dạng: $1, i, a$: có nghĩa điều chỉnh hiệu năng tính toán của máy tính i thành a ($T_i = a, a \leq 10^4$).

Phép kiểm tra thứ hai có dạng: $2, i$: Tính tổng hiệu năng tính toán của tất cả các máy tính thuộc cây con gốc i (gốc là máy tính i).

Dữ liệu: Vào từ file văn bản TreeComputer.Inp gồm:

- Dòng đầu ghi hai số nguyên dương n, q ($q, n \leq 10^5$).



- Dòng thứ hai ghi n số nguyên dương T_1, T_2, \dots, T_n ($T_i \leq 10^4$) là hiệu năng tính toán của n máy tính.
- $n - 1$ dòng cuối tiếp theo, mỗi dòng ghi hai số nguyên u, v mô tả một cặp nối giữa hai máy tính u và v .
- q dòng cuối, mỗi dòng ghi một phép kiểm tra thuộc một trong hai phép kiểm tra trên.

Kết quả: Đưa ra file TreeComputer.Out là kết quả của các phép kiểm tra loại 2, mỗi kết quả ghi trên một dòng (lần lượt theo thứ tự trong file input).

Ví dụ:

TreeComputer.Inp	TreeComputer.Out	Hình minh họa
5 4 1 2 3 1 4 1 2 1 3 3 4 3 5 2 1 2 3 1 5 6 2 3	11 8 10	