A. ¿Qué aristas van en un AGM?

2 s, 256 MB

Un día Echu y Tuki estaban aburridos y se pusieron a conversar sobre árboles generadores mínimos (AGMs). Tuki le recordaba a Echu que dado un grafo conexo siempre existe un árbol generador mínimo, pero que no es necesariamente único, pueden existir varios.

Echu al escuchar esas palabras se puso a reflexionar y se preguntó si para cada arista de un grafo conexo, se podía ver si la arista pertenecía a **todos** los árboles generadores mínimos que existen, a **al menos un** árbol generador mínimo, o a **ninguno**.

Tuki se quedó pensando y le respondió que si tiene un grafo con todas aristas de distinto peso, entonces existe un sólo árbol generador mínimo. Pueden existir varios árboles generadores mínimos solamente si existen múltiples aristas con pesos iguales, ya que en algunos casos una arista se puede reemplazar por otra del mismo peso en un AGM.

Echu pensó rápidamente que cuando se topaba con un grupo de aristas del mismo peso quizás se podía construir un nuevo grafo G' cuando se estuviese construyendo el AGM, donde cada vértice sea una componente conexa del bosque original construido por el AGM y las aristas que se quieren analizar (las que son del mismo peso) son agregadas para conectar las componentes. Pensó que eso podría ser útil para determinar la clasificación de las aristas. Ver si hay aristas que son puentes o se genera algún ciclo puede ayudar.

Después de discutir un rato, a Tuki le pareció genial el problema que planteó Echu y decidió sugerirle a Eric que agregue ese problema al TP2 de la materia. Tu tarea ahora es resolver el problema planteado por Echu.

Input

La primera línea contiene dos enteros n y m ($2 \leq n \leq 10^5$, $n-1 \leq m \leq min(10^5, \frac{n(n-1)}{2})$) — El número de vértices y aristas del grafo, respectivamente. Luego, siguen m líneas, cada una con 3 enteros — se describe a las aristas del grafo como "a_i b_i w_i" ($1 \leq a_i, b_i \leq n, 1 \leq w_i \leq 10^6, a_i \neq b_i$), donde a_i y b_i son los números de vértices conectados por la i-ésima arista, y w_i es el peso de la i-ésima arista. Está garantizado que el grafo es conexo y que no contiene loops ni múltiples aristas.

Output

Imprimir m líneas — la respuesta para todas las aristas. Si la i-ésima arista está incluida en todos los AGM, imprimir "`any`". Si la i-ésima arista está incluida en al menos un AGM, imprimir "`at least one`". Si la i-ésima arista no está incluida en ningún AGM, imprimir "`none`". Imprimir las respuestas para las aristas en el orden en el cual las aristas fueron especificadas en el input.

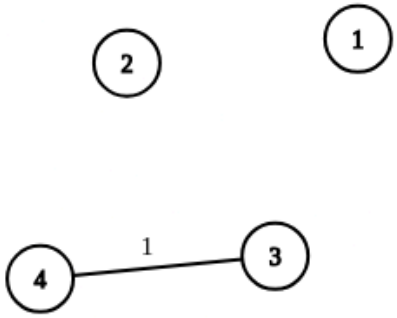
input
4 5 1 2 101 1 3 100 2 3 2 2 4 2 3 4 1

output
none any at least one at least one any

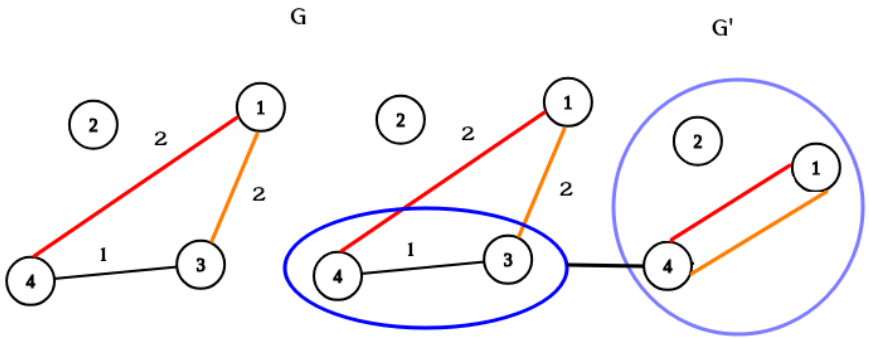
input
3 3 1 2 1 2 3 1 1 3 2
output
any any none

input
3 3 1 2 1 2 3 1 1 3 1
output
at least one at least one at least one

En el primer ejemplo, podemos verlo paso a paso. Primero se agrega la arista de menor peso.



Luego, se agregan dos aristas que tienen el mismo peso.



Para determinar si están en al menos un AGM, se construye el grafo G' de componentes conexas. Se agregan las aristas de peso dos y vemos que allí se produce un ciclo. Esto quiere decir que podría construir para cada arista, un AGM distinto. Se agrega la arista de peso 100, como no forma un ciclo, entonces se puede agregar sin problemas. Por ultimo, al agregar la arista de peso 101, se formaría un ciclo en el grafo G, con lo cual significa que la podemos descartar y no pertenece a ningún AGM.

En el segundo ejemplo, el AGM a generar es único, y contiene a las primeras dos aristas.

En el tercer ejemplo, podemos ver que tenemos un grafo con un ciclo de 3 vértices. Podemos generar 3 AGMs, donde podemos excluir alguna arista para cada uno.

B. Igna, Martín y la bipartitud

2 s., 256 MB

Igna y Martín continúan sus aventuras! Como todos en TDA saben, a los JTPs les gustan los grafos bipartitos, especialmente los árboles.

Un grafo bipartito es un grafo cuyos vértices se pueden dividir en 2 conjuntos de tal manera que para cada arista (u, v) que pertenece al grafo, u y v pertenecen a diferentes conjuntos. Podés encontrar una definición más formal de un grafo bipartito en la sección de notas a continuación.

Los JTPs le dieron a Igna y Martín un árbol que consiste en n nodos y les pidieron que agregaran aristas al mismo de tal manera que el grafo siga siendo bipartito. Además, después de agregar estas aristas, el grafo debe ser simple (no debe contener loops, ni aristas múltiples). ¿Cuál es el número máximo de aristas que pueden agregar?

Un loop es una arista que conecta a un nodo consigo mismo. Un grafo no contiene aristas múltiples cuando para cada par de nodos no hay más de una arista entre ellos. Un ciclo y un loop **no son lo mismo**.

Input

La primera línea de entrada contiene un entero n — el número de nodos en el árbol ($1 \leq n \leq 10^5$).

Las siguientes $n - 1$ líneas contienen enteros u y v ($1 \leq u, v \leq n$, $u \neq v$) — la descripción de las aristas del árbol.

Se garantiza que el grafo dado es un árbol.

Output

Imprimir un número entero — el número máximo de aristas que Igna y Martín pueden agregar al árbol mientras cumplan con las condiciones.

input
3 1 2 1 3
output
0

input
5 1 2 2 3 3 4 4 5
output
2

Definición de árbol: https://en.wikipedia.org/wiki/Tree_(graph_theory)

Definición de grafo bipartito: https://en.wikipedia.org/wiki/Bipartite_graph

En el primer caso de prueba, la única arista que se puede agregar de tal manera que el grafo no contenga bucles ni aristas múltiples es $(2, 3)$, pero agregar esta arista haría que el grafo deje de ser bipartito, por lo que la respuesta es 0.

En el segundo caso de prueba, Igna y Martín pueden agregar las aristas $(1, 4)$ y $(2, 5)$.

C. Juli y los túneles de Exactas

3 s., 256 MB

En estos últimos años, Juli estuvo muy ocupado estudiando para exámenes y el concurso de Ay1. Ahora que ya se recibió quiere relajarse y recorrer todo Exactas tranquilo.

Exactas consta de n aulas numeradas del 1 al n. Juli comienza a caminar desde el aula número 1 (en particular, el aula 6 del Pabe II) y sigue alguna secuencia de aulas. Caminar desde el espacio número i hasta otro espacio j requiere $|i - j|$ unidades de energía. La *energía total* gastada por Juli al visitar una secuencia de aulas $e_1 = 1, e_2, \ldots, e_k$ es igual a $\sum_{i=1}^{k-1} |e_i - e_{i+1}|$ unidades de energía.

Por supuesto, caminar sería aburrido si no hubiera atajos. Un *atajo* es un túnel oculto que permite a Juli caminar de un aula a otra requiriendo solo 1 unidad de energía. Hay exactamente n atajos en Exactas; el i-ésimo de ellos permite caminar desde el aula i hasta el aula a_i ($i \leq a_i \leq a_{i+1}$) (pero no en la dirección opuesta), por lo tanto, hay exactamente un atajo que comienza en cada aula. Formalmente, si Juli elige una secuencia $e_1 = 1, e_2, \ldots, e_k$, entonces para cada $1 \leq i < k$ que satisface $e_{i+1} = a_{e_i}$ y $a_{e_i} \neq e_i$, Juli gastará **solo 1 unidad de energía** en lugar de $|e_i - e_{i+1}|$ al caminar desde el aula e_i hasta el aula e_{i+1}. Por ejemplo, si Juli elige una secuencia $e_1 = 1, e_2 = a_{e_1}, e_3 = a_{e_2}, \ldots, e_k = a_{e_{k-1}}$, gastará exactamente $k - 1$ unidades de energía total al recorrerlas.

Antes de emprender su aventura, Juli te pide que encuentres la cantidad mínima de energía requerida para alcanzar cada una de las intersecciones desde el aula inicial. Formalmente, para cada $1 \leq i \leq n$, Juli está interesado en encontrar la mínima energía total posible de alguna secuencia $e_1 = 1, e_2, \ldots, e_k = i$.

Input

La primera línea contiene un entero n $(1 \leq n \leq 200,000)$ — el número de aulas en Exactas (cambian cada cuatri).

La segunda línea contiene n enteros $a_1, a_2, \ldots, a_n$ $(i \leq a_i \leq n, a_i \leq a_{i+1} \forall i < n)$, describiendo los atajos de Exactas, que permiten caminar desde el aula i hasta el aula a_i usando solo 1 unidad de energía. Tené en cuenta que los atajos no permiten caminar en dirección opuesta (de a_i a i).

Output

En la única línea, imprimí n enteros $m_1, m_2, \ldots, m_n$, donde m_i denota la menor cantidad de energía total requerida para caminar desde el aula 1 hasta el aula i.

input
3 2 2 3

output
0 1 2

input
5 1 2 3 4 5
output
0 1 2 3 4

input
7 4 4 4 4 7 7 7
output
0 1 2 1 2 3 3

En el primer caso de muestra, las secuencias deseadas son:

$1 : (1); m_1 = 0;$

$2 : (1, 2); m_2 = 1;$

$3 : (1, 3); m_3 = |3 - 1| = 2.$

En el segundo caso de muestra, la secuencia para cualquier aula $1 < i$ es siempre $(1, i)$ y $m_i = |1 - i|$.

En el tercer caso de muestra, unas secuencias posibles son:

$1 : (1); m_1 = 0;$

$2 : (1, 2); m_2 = |2 - 1| = 1;$

$3 : (1, 4, 3); m_3 = 1 + |4 - 3| = 2;$

$4 : (1, 4); m_4 = 1;$

$5 : (1, 4, 5); m_5 = 1 + |4 - 5| = 2;$

$6 : (1, 4, 6); m_6 = 1 + |4 - 6| = 3;$

$7 : (1, 4, 5, 7); m_7 = 1 + |4 - 5| + 1 = 3$