



> A - ¿Es un árbol?

Se te da un árbol sin pesos y no dirigido, escribe un programa para verificar si es un árbol.

Un árbol está definido como un grafo que no tiene ciclos y todo par de nodos sólo tiene un camino posible.

* Input:

La primera línea de la entrada contiene dos enteros separados por espacios, N y M : número de nodos y número de aristas en el grafo ($0 \leq N \leq 10000, 0 \leq M \leq 20000$).

Las siguientes M líneas contienen las M aristas del grafo - Cada línea contiene un par (u, v) que indica que hay una arista entre el nodo u y el nodo v ($1 \leq u, v \leq N$).

* Output:

Muestre por pantalla YES si el grafo es un árbol, de lo contrario, Muestre por pantalla NO.

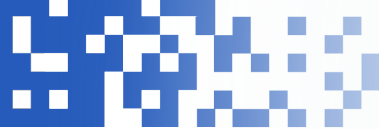
* Ejemplo

Input:

```
3 2
1 2
2 3
```

Output:

```
YES
```



> B - Camino más largo en un árbol

Se te da un árbol sin pesos y no dirigido, escribe un programa para mostrar la longitud del camino más largo (de un nodo a otro) de ese árbol.

La longitud de un camino en este caso es el número de aristas que recorreremos desde el origen hasta el destino.

Un árbol está definido como un grafo que no tiene ciclos y todo par de nodos sólo tiene un camino posible.

* Input:

La primera línea de la entrada contiene un entero N - número de nodos en el árbol ($0 < N \leq 10000$). Las siguientes $N - 1$ líneas contienen $N - 1$ aristas del árbol - Cada línea contiene un par (u, v) que indica que hay una arista entre el nodo u y el nodo v ($1 \leq u, v \leq N$).

* Output:

Muestre por pantalla el largo del camino más largo

* Ejemplo

Input:

```
3
1 2
2 3
```

Output:

```
2
```



> C - Rumor

Javier se prometió a sí mismo que nunca jugaría videojuegos... Pero recientemente, OCIGames - una conocida empresa de desarrollo de juegos - publicó su juego más reciente, OCICraft, y se volvió realmente popular. Por supuesto, Javier comenzó a jugarlo.

Ahora intenta resolver una misión. La tarea es llegar a un asentamiento llamado OCI-City y difundir un rumor en él.

Javier sabe que hay n personajes en OCI-City. Algunos personajes son amigos entre sí y comparten la información que obtienen. Además, Javier sabe que puede sobornar a cada personaje para que comience a difundir el rumor; el i -ésimo personaje quiere c_i oro a cambio de difundir el rumor. Cuando un personaje escucha el rumor, se lo cuenta a todos sus amigos, y ellos comienzan a difundir el rumor a sus amigos (gratis), y así sucesivamente.

La misión se completa cuando todos los n personajes conocen el rumor. ¿Cuál es la cantidad mínima de oro que Javier necesita gastar para completar la misión?

* Input:

La primera línea contiene dos números enteros n y m ($1 \leq n \leq 105, 0 \leq m \leq 105$) - el número de personajes en OCI-City y el número de pares de amigos.

La segunda línea contiene n números enteros c_i ($0 \leq c_i \leq 10^9$) - la cantidad de oro que el i -ésimo personaje pide para comenzar a difundir el rumor.

Luego siguen m líneas, cada una conteniendo un par de números (x_i, y_i) que representan que los personajes x_i y y_i son amigos ($1 \leq x_i, y_i \leq n, x_i \neq y_i$). Se garantiza que cada par se muestra como máximo una vez.

* Output:

Muestre por pantalla un número - la cantidad mínima de oro que Javier tiene que gastar para completar la misión.

* Ejemplo

Input:

```
5 2
2 5 3 4 8
1 4
4 5
```

Output:

```
10
```

Input:

```
10 0
1 2 3 4 5 6 7 8 9 10
```

Output:

```
55
```



Input:

```
10 5
1 6 2 7 3 8 4 9 5 10
1 2
3 4
5 6
7 8
9 10
```

Output:

```
15
```

Nota:

En el primer ejemplo, la mejor decisión es sobornar al primer personaje (él difundirá el rumor al cuarto personaje, y el cuarto lo difundirá al quinto). También Javier tiene que sobornar al segundo y al tercer personaje, para que ellos conozcan el rumor.

En el segundo ejemplo, Javier tiene que sobornar a todos.

En el tercer ejemplo, la decisión óptima es sobornar al primero, al tercero, al quinto, al séptimo y al noveno personaje.