

Numbering (numbering)

Dado un bosque de N vértices, una numeración es una asignación de números positivos a cada arista del bosque. Se dice que una numeración es bonita si, para cada vértice, las aristas tienen los números $1, 2, \dots, d$ en algún orden (donde d es el grado del vértice).

Tienes N enteros positivos A_0, \dots, A_{N-1} . Determina si existe un bosque en N vértices que cumpla:

- para cada $0 \leq i \leq N - 1$, el grado del vértice i es A_i ;
- admite una numeración bonita.

Adicionalmente si existe dicho bosque, construye un ejemplo.

Implementación

Tienes que mandar un solo archivo `.cpp`.

📁 Entre los adjuntos de esta tarea, tendrás un archivo template `numbering.cpp` con un ejemplo de implementación.

Tienes que implementar la siguiente función:

```
C++    variant<bool, vector<pair<int, int>>> find_numbering(int N, vector<int>
      A);
```

- Entero N representando el número de vértices.
- Un vector A , indexado de 0 a $N - 1$, conteniendo los valores A_0, A_1, \dots, A_{N-1} , donde A_i es el grado del vértice i .
- La función tiene que devolver un booleano o un vector de pares de enteros.
 - Si no existe un bosque válido (que satisfaga las condiciones del enunciado), la función debe devolver `false`.
 - Si existe un bosque válido, tienes dos opciones:
 - Para recibir la puntuación entera, la función debe devolver un vector de pares de enteros, representando un bosque válido.
 - Para recibir puntuación parcial, debe devolver `true` o un vector que no describa un bosque válido.

El grader llamará la función `find_numbering` e imprimirá el siguiente archivo de salida:

- Si el valor de retorno es `false`, imprimirá una sola línea con la palabra `NO`.
- Si el valor de retorno es `true`, imprimirá una sola línea con la palabra `YES`.
- Si el valor de retorno es un vector de pares de enteros de longitud M , imprimirá una línea con la palabra `YES`, seguido de una línea con M , seguido de M líneas con los pares de los vectores.

Sample Grader

En el directorio de la tarea hay una versión simplificada del grader, que puedes usar para ejecutar tu solución en local. El grader simplificado lee el input de `stdin`, llama las funciones que tienes que implementar, y finalmente escribe la salida en `stdout`.

La entrada está formada por 2 líneas, que contienen:

- Línea 1: el entero N .
- Línea 2: A_0, A_1, \dots, A_{N-1} .

La salida está hecha de múltiples líneas, conteniendo los valores devueltos por la función `find_numbering`.

Restricciones

- $2 \leq N \leq 10^5$.
- $0 \leq A_i \leq N - 1$.

Puntuación

Tu programa será probado con casos de pruebas agrupados en subtareas. La puntuación de cada subtarea será la mínima puntuación obtenida en cada juego de pruebas.

- **Subtask 1 [0 puntos]**: Sample test cases.
- **Subtask 2 [16 puntos]**: $A_i \leq 2$.
- **Subtask 3 [12 puntos]**: $A_i \leq 3$.
- **Subtask 4 [16 puntos]**: $N \leq 12$.
- **Subtask 5 [10 puntos]**: $N \leq 500$.
- **Subtask 6 [24 puntos]**: Sin restricciones adicionales.

Para cada caso de pruebas en que exista una solución válida, tu solución:

- recibe puntuación completa si devuelve un bosque válido.
- tiene 50 % si devuelve `true` o un vector que no describa un bosque válido.
- recibe 0 puntos alternativamente.

Para cada juego de pruebas en que no exista una solución válida, tu solución:

- recibe puntuación total si devuelve `false`.
- recibe 0 puntos alternativamente.

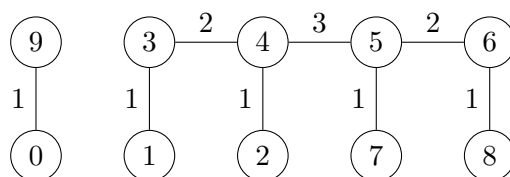
Ejemplos de entrada/salida

stdin	stdout
4 1 1 2 1	NO
10 1 1 1 2 3 3 2 1 1 1	YES 8 0 9 1 3 2 4 3 4 4 5 5 6 5 7 6 8

Explicación

En el **primer caso de prueba**, queremos un bosque válido con 4 vertices: 3 con grado 1 y 1 con grado 2. Podemos demostrar que esto no es posible. Imagina que existe un bosque así. Entonces queremos que haya una arista con el número 2 que salga del vértice de grado 2. Esta arista tiene que conectar con otro nodo que tenga grado como mínimo 2. Pero, tal vértice no existe, ya que todos los otros vértices tienen grado 1.

En el **segundo juego de pruebas**, queremos un bosque válido con 10 nodos: 6 con grado 1, 2 con grado 2 y 2 con grado 3. Este bosque si que existe y se muestra a continuación:



Fíjate que los nodos 4 y 5 tienen tres aristas 1, 2 y 3.

Además los vértices 3 y 6 tienen dos aristas con los números 1 y 2.

Finalmente los vértices 0, 1, 2, 7, 8 y 9 tienen una arista con el número 1.