

Numbering (numbering)

Dada uma floresta com N nós, uma numeração da mesma é uma atribuição de números inteiros positivos a cada aresta (ligação) da floresta. Uma numeração é bonita se, para cada nó, as suas arestas tiverem os números $1, 2, \dots, d$ numa qualquer ordem (onde d é o grau do nó).

São-te dados N inteiros positivos A_0, \dots, A_{N-1} . Determina se existe uma floresta em N nós tal que:

- para $0 \leq i \leq N - 1$, o grau do nó i é A_i ;
- admite pelo menos uma numeração bonita.

Além disso, se existir uma floresta deste tipo, deves construir um exemplo.

Implementação

Deves submeter um único ficheiro de código `.cpp`.

🔍 Entre os ficheiros do problema encontrarás um template `numbering.cpp` com um exemplo de implementação.

Tens de implementar a seguinte função:

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

- O inteiro N representa o número de nós.
- O array A , indexado de 0 até $N - 1$, contém os valores A_0, A_1, \dots, A_{N-1} , onde A_i é o grau do i -ésimo nó.
- A função deve devolver um valor booleano ou um array de pares de inteiros.
 - Se não existir nenhuma floresta válida (satisfazendo as condições do enunciado), a função deve devolver `false`.
 - Se existir uma floresta válida, tens duas opções:
 - * Para receber a pontuação máxima, a função deve devolver um array de pares de inteiros, representando uma floresta válida.
 - * Para receber uma pontuação parcial, a função deve devolver `true` ou um qualquer array de inteiros que não descreva uma floresta válida.

O avaliador irá chamar a função `find_numbering` e irá escrever o seguinte no ficheiro de output:

- Se o valor devolvido for `false`, irá escrever uma única linha com a string `NO`.
- Se o valor devolvido for `true`, irá escrever uma única linha com a string `YES`.
- Se o valor devolvido for um array de pares de inteiros com tamanho M , irá escrever uma linha com a string `YES`, seguida de uma linha com M , seguida de M linhas com os pares do array.

Avaliador Padrão

O diretório do problema contém uma versão simplificada do avaliador oficial, que podes usar para testar o teu problema localmente. O avaliador exemplo lê os dados de input de `stdin`, chama as funções que deves implementar, e finalmente escreve o output para `stdout`.

O input é feito de 2 linhas, contendo:

- Linha 1: o inteiro N .
- Linha 2: A_0, A_1, \dots, A_{N-1} .

O output é feito de múltiplas linhas, contendo os valores devolvidos pela função `find_numbering`.

Restrições

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

Pontuação

O teu programa será testado num conjunto de casos de teste agrupados por subtarefa. A pontuação de uma subtarefa será o mínimo das pontuações obtidas em cada um dos casos de teste.

- **Subtarefa 1 [0 pontos]:** Casos de exemplo.
- **Subtarefa 2 [16 pontos]:** $A_i \leq 2$.
- **Subtarefa 3 [12 pontos]:** $A_i \leq 3$.
- **Subtarefa 4 [16 pontos]:** Seja $\text{count}(i)$ o número de ocorrências de i em A . É garantido que $\text{count}(i) \geq \text{count}(i+1) + \text{count}(i+2) + \dots$ para $1 \leq i \leq N-1$.
- **Subtarefa 5 [10 pontos]:** $N \leq 12$.
- **Subtarefa 6 [24 pontos]:** $N \leq 500$.
- **Subtarefa 7 [22 pontos]:** Nenhuma restrição adicional.

Para cada caso de teste em que exista uma floresta válida, a tua solução:

- obtém pontuação máxima se devolver uma floresta válida.
- obtém 50% dos pontos se devolver `true` ou um array que não descreva uma floresta válida.
- obtém 0 pontos caso contrário.

Para cada caso de teste em que não exista uma floresta válida, a tua solução:

- obtém pontuação máxima se devolver `false`.
- obtém 0 pontos caso contrário.

Exemplos

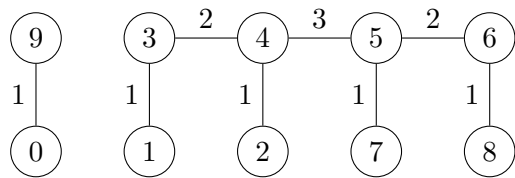
stdin	stdout
4 1 1 2 1	NO
10 1 1 1 2 3 3 2 1 1 1	YES 8 0 9 1 3 2 4 3 4 4 5 5 6 5 7 6 8

Explicação

No **primeiro caso de exemplo**, queremos uma floresta válida com 4 nós: 3 com grau 1 e 1 com grau 2. Podemos mostrar que isto não é possível. Supõe um

No **primeiro caso de exemplo**, queremos uma floresta válida com 4 nós: 3 com grau 1 e 1 com grau 2. Podemos mostrar que isto não é possível. Suponhamos que essa floresta existe, então deve haver uma aresta com o número 2 a partir do nó com grau 2. Esta aresta liga-se a outro nó que deve ter grau pelo menos 2. No entanto, esse nó não existe, uma vez que todos os outros nós têm grau 1.

No **segundo caso de exemplo**, queremos uma floresta válida com 10 nós: 6 com grau 1, 2 com grau 2 e 2 com grau 3. Essa floresta existe e o resultado é apresentado abaixo:



Nota que os nós 4 e 5 têm três arestas numerados com 1, 2 e 3.

Além disso, os nós 3 e 6 têm duas arestas numeradas com 1 e 2.

Finalmente, os nós 0, 1, 2, 7, 8 e 9 têm uma aresta numerada com 1.