

## Numbering (numbering)

Gegeven een bos met  $N$  knopen, een nummering daarvan is een toekenning van positieve gehele getallen aan elke kant van het bos. Een nummering is mooi als, voor elke knoop, de kanten de nummers  $1, 2, \dots, d$  bevat in een willekeurige volgorde (waar  $d$  de graad van de knoop).


Je krijgt  $N$  positieve integers  $A_0, \dots, A_{N-1}$ . Bepaal of er een bos bestaat met  $N$  knopen zodat:

- voor elke  $0 \leq i \leq N - 1$ , de graad van knoop  $i$  is  $A_i$ ;
- het heeft minstens één mooie nummering.

Verder, als er zo'n bos bestaat, geef een voorbeeld.

## Implementatie

Je moet één .cpp-bestand inleveren.

 In de bijlagen van deze opgave vind je een sjabloon `numbering.cpp` met een voorbeeld-implementatie.

Je moet de volgende functie implementeren:

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

- Integer  $N$  die staat voor het aantal knopen.
- De array  $A$ , genummerd van 0 tot en met  $N - 1$ , bevat de getallen  $A_0, A_1, \dots, A_{N-1}$ , waarbij  $A_i$  de graad is van knoop  $i$ .
- The function should return either a boolean or an array of pairs of integers.
  - Als er geen geldig (voldoet aan de eisen in de omschrijving) bos bestaat, moet de functie **false** teruggeven.
  - Als er een geldig bos bestaat heb je twee opties:
    - \* Om volle punten te krijgen moet je functie een array van paren integers teruggeven, die staat voor de kanten van een geldig bos.
    - \* Om een gedeeltelijke score te krijgen moet je functie **true** teruggeven, of een array met paren integers die niet een geldig bos beschrijft.

De grader roept de functie `find_numbering` aan en zal het volgende schrijven naar het uitvoerbestand:

- Als je functie **false** teruggeeft, schrijft het een enkele regel met de tekst NO.
- Als je functie **true** teruggeeft, schrijft het een enkele regel met de tekst YES.
- Als je functie een array met paren integers teruggeeft van lengte  $M$ , schrijft het één regel met de tekst YES, gevolgd door één regel met  $M$ , gevolgd door  $M$  regels met de paren van de array.

## Voorbeeld Grader

De map van de opdracht bevat een voorbeeldgrader, een versimpelde versie van de jury's grader, die je kan gebruiken om je oplossing lokaal te testen. De voorbeeldgrader leest de invoer van `stdin`, roept de functies aan die je moet implementeren en schrijft vervolgens de uitvoer naar `stdout`.

De invoer bestaat uit 2 regels met:

- Regel 1: het integer  $N$ .
- Regel 2:  $A_0, A_1, \dots, A_{N-1}$ .

De uitvoer bestaat uit meerdere regels die de waarden bevat die door de functie `find_numbering` wordt teruggegeven.

## Randvoorwaarden

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

## Scoring

Je programma zal worden getest op een set van testgevallen die per deelopgaven zijn gegroepeerd. De score voor een deelopgave is het minimum van de scores voor de testgevallen.

- **Subtask 1 [ 0 punten]**: Voorbeeld testgevallen.
- **Subtask 2 [16 punten]**:  $A_i \leq 2$ .
- **Subtask 3 [12 punten]**:  $A_i \leq 3$ .
- **Subtask 4 [16 punten]**: Laat `aantal( $i$ )` het aantal keren dat  $i$  voorkomt in  $A$ . Het is gegarandeerd dat `count( $i$ )`  $\geq$  `count( $i + 1$ )` + `count( $i + 2$ )` + ... voor alle  $1 \leq i \leq N - 1$ .
- **Subtask 5 [10 punten]**:  $N \leq 12$ .
- **Subtask 6 [24 punten]**:  $N \leq 500$ .
- **Subtask 7 [22 punten]**: Geen extra randvoorwaarden.

Voor elk testgeval waarin een geldig bos bestaat krijgt je oplossing:

- volle punten als het een geldig bos teruggeeft.
- 50% van de punten als het `true` of een array dat geen geldig bos omschrijft teruggeeft.
- 0 punten in alle andere gevallen.

Voor elk testgeval waarin geen geldig bos bestaat krijgt je oplossing:

- volle punten als het `false` teruggeeft.
- 0 punten in alle andere gevallen.

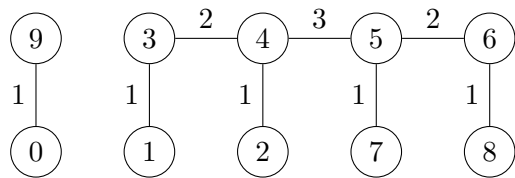
# Voorbeelden

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

## Uitleg

In het **eerste voorbeeld**, willen we een geldig bos met 4 knopen: 2 met graad 1 en 1 met graad 2. We kunnen aantonen dat dit niet kan. Stel dat zo'n bos wel bestaat, dan is er een kant met nummer 2 vanuit de knoop met graad 2. Deze kant moet verbinden met een andere knoop met graad 2. Echter, zo'n knoop bestaat niet aangezien alle andere knopen graad 1 hebben.

In het **tweede voorbeeld**, willen we een geldig bos met 10 knopen: 6 met graad 1, 2 met graad 2 en 2 met graad 3. Zo'n bos bestaat en de uitvoer is hieronder afgebeeld:



Merk op dat knopen 4 en 5 drie kanten hebben gelabeld met 1, 2 en 3. Verder, knopen 3 en 6 hebben twee kanten gelabeld met 1 en 2. Als laatste, knopen 0, 1, 2, 7, 8 en 9 hebben één kant gelabeld 1.