

Отборност

Време: 4 секунди

Памет: 1024 MiB

Антон наскоро се разхождал из Каунас по време на EJOI, когато открил N места, на които участниците се събират. Местата са номерирани от 0 до $N - 1$. Във всяко място има членове на точно един отбор и отборите са номерирани от 0 до $K - 1$. Обърнете внимание, че членове на един и същ отбор могат да заемат **произволно много** места. Някои отбори може да не заемат нито едно място.

Местата са свързани с $N - 1$ двупосочни пътя, така че между всеки две места съществува точно един *прост път*, следователно те образуват дърво. Припомняме, че прост път между две места е последователност от различни места, в която всеки две последователни места са свързани с път. *Дължината* на един път е броят на пътищата, които той използва, което е с едно по-малко от броя на местата, през които преминава.

Антон иска да се разходи по прост път, като посети възможно най-много места. Един прост път се нарича *интересен*, ако дължината му е максималната възможна сред всички прости пътища в дървото. Отборността на един път е броят на различните отбори, които Антон среща по него.

Вашата задача е да намерите сумата от отборностите на всички различни интересни пътища. Два интересни пътя се считат за еднакви тогава и само тогава, когато посещават точно едно и също множество от места (в частност, обхождането на един път в обратна посока не създава различен път).

Детайли по имплементацията

Трябва да имплементирате следната функция:

```
long long teamfulness(int N, int K, std::vector<int> a,  
                      std::vector<int> u, std::vector<int> v)
```

- N : броят на местата;
- K : броят на отборите;
- a : вектор от N цели числа, където a_i е отборът, заемащ място i , за всяко $0 \leq i < N$;

- u, v : вектори от $N - 1$ цели числа, където u_i и v_i са двете места, свързани от i -тия път, за всяко $0 \leq i < N - 1$.

Тази функция се извиква точно веднъж за всеки тест и трябва да върне сумата от отборностите на всички интересни пътища.

Ограничения

- $3 \leq N \leq 10^6$
- $1 \leq K \leq N$
- $0 \leq a_i < K$ за всяко $0 \leq i < N$
- $0 \leq u_i, v_i < N$ за всяко $0 \leq i < N - 1$

Примери

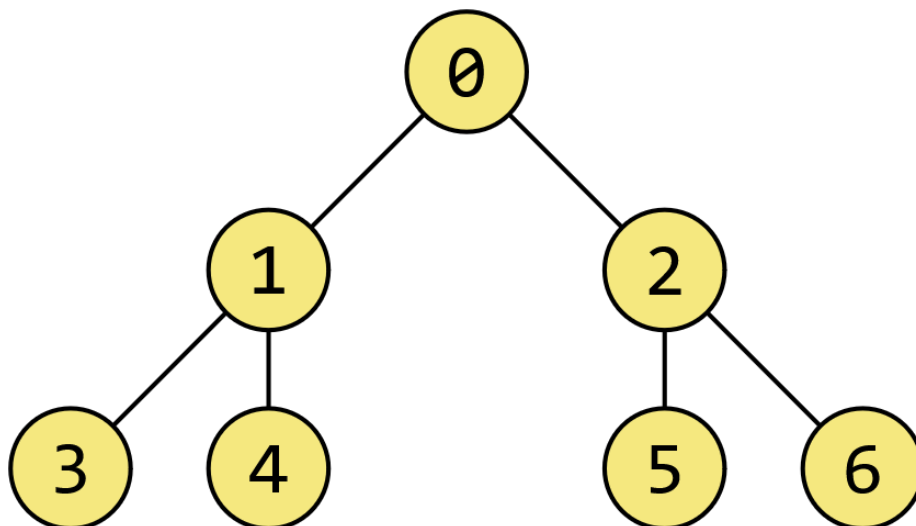
Примерен вход за грейдъра	Примерен изход на грейдъра
6 3 1 0 0 1 2 1 0 1 0 2 0 3 0 4 0 5	21
7 1 0 0 0 0 0 0 0 0 1 0 2 1 3 1 4 2 5 2 6	4
6 3 0 1 2 0 1 2 0 1 1 2 2 3 1 4 2 5	11

Обяснение на пример 1

Максималната дължина на прост път е 2 (2 пътя и 3 места), следователно интересните пътища ще имат дължина 2. Има два интересни пътя с отборност 3, седем интересни пътя с отборност 2 и един интересен път с отборност 1, което дава обща сума 21.

Обяснение на пример 2

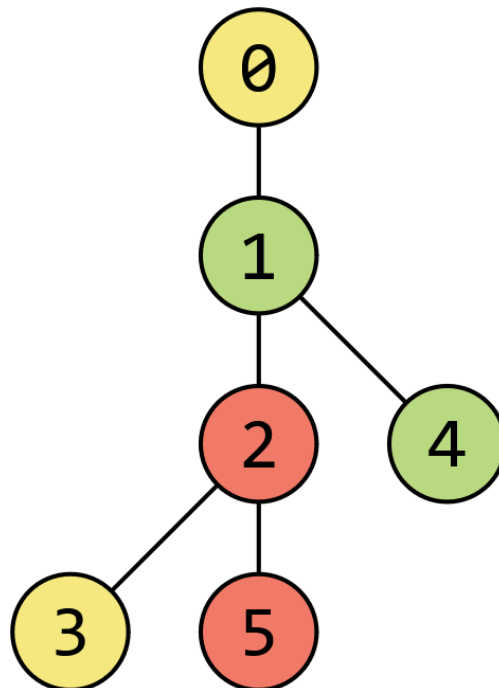
Местата и пътищата, които ги свързват, са изобразени по следния начин (отбор 0 е оцветен в жълто):



Отборността на всеки път е 1, тъй като има само един отбор (отбор 0), чиито членове се събират на всяко място. Има 4 интересни пътя (с дължина 4), следователно сумата от отборностите на всички интересни пътища също е 4.

Обяснение на пример 3

Местата и пътищата, които ги свързват, са изобразени по следния начин (отбор 0 е оцветен в жълто, отбор 1 — в зелено, а отбор 2 — в червено):



Интересните пътища имат дължина 3. Общо има четири интересни пътя: три от тях имат отборност 3, а един има отборност 2. Следователно общата сума от отборностите на всички интересни пътища е 11.

Подзадачи

Подзадача	Точки	N	K	Допълнителни ограничения
0	0	-	-	Примерите.
1	4	$\leq 10^6$	$\leq N$	Всяко място е пряко свързано с най-много 2 други места.
2	7			Съществува място, което е пряко свързано с всяко друго място.
3	9	≤ 200	$\leq N$	-
4	10	$\leq 2 \cdot 10^3$		
5	10	$\leq 10^6$	$= 1$	
6	9		≤ 2	
7	11	$\leq 2 \cdot 10^5$	≤ 50	
8	12		$\leq N$	
9	13	$\leq 10^6$	$\leq N$	Дължината на интересния път е нечетна.
10	15			-

Примерен грейдър

Входният формат е следният:

- ред 1: две цели числа — стойностите на N и K ;
- ред 2: N цели числа $a_0, a_1, \dots, a_{N-1}$, където a_i е отборът, заемащ място i ;
- ред $3 + i$: две цели числа u и v — двете места, свързани от i -тия път.

Изходният формат е следният:

- ред 1: едно цяло число — стойността, върната от извикването на функцията.