

C. みんなでコーディング (Team Coding)

問題名	みんなでコーディング (Team Coding)
実行時間制限	4 秒
メモリ制限	1 GB

Eindhoven Gigantic Open-Source Institute (EGOI) という会社は非常に階層的な構造からなっている。CEO の Anneke を除き、社内の $N - 1$ 人の従業員にはそれぞれ報告先となる 1 人の上司がいる。EGOI 社の階層構造にはサイクルがなく、社内の階層構造は Anneke に対応する頂点を根とする根付き木とみなすことができる。この会社は多様性に富んでおり、社員は K 種類のプログラミング言語を用いてコードを書くが、どの社員も好きなプログラミング言語をちょうど 1 つだけ持っている。

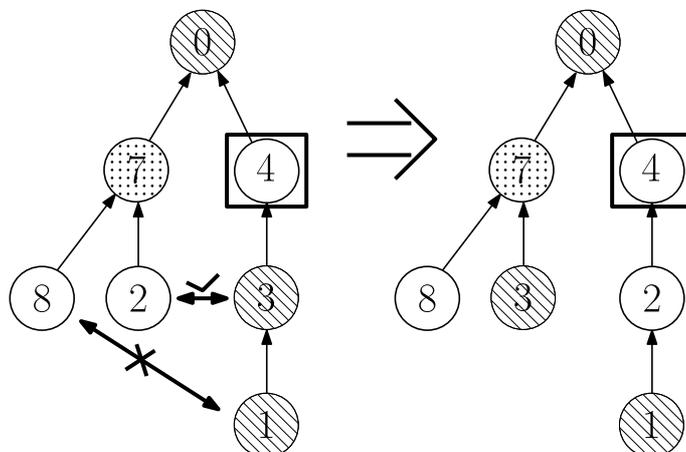
Anneke は社内の 1 つのチームで取り組むべき大きな新プロジェクトを抱えている。彼女はこのプロジェクトにできるだけ多くのリソースを投入したい。このプロジェクトに取り組むチームを決めるために、彼女は次の作業を行うことにした。

1. チームのリーダーを選ぶ。これにより、プロジェクト内のコードに利用されるプログラミング言語が決定される。社内の階層構造において、チームリーダーの配下（チームリーダーを根とする部分木内）に属する社員のうち、チームリーダーと同じプログラミング言語を好む社員全員がこのプロジェクトに取り組む。
2. チームリーダーと同じプログラミング言語を好む社員を後述の入れ替え操作を行ってチーム内に入れ、プロジェクトに従事する社員の数を増やす。

ここで、プロジェクトに従事する社員数を最大化するために、Anneke は以下の入れ替え操作を好きな回数行うことができる。

1. まず、以下の条件を満たす 2 人の社員を選ぶ。
 - 一方の社員は、現在チームリーダーの配下にいるが、チームリーダーとは異なるプログラミング言語を好んでいる。
 - もう一方の社員は、現在チームリーダーの配下にはいないが、チームリーダーと同じプログラミング言語を好んでいる。さらに、社内の階層構造において、この社員ともう一方の社員は同じ階層にいる。つまり、Anneke に報告するまでに中継する上司の人数は同じである。社内の階層構造を木構造とみなすと、これは 2 人の社員に対応する頂点の深さが等しいことを意味する。

2. 社内の階層構造において、選んだ2人の社員の立ち位置を入れ替える（その他の社員の立ち位置は変化しない）。特に、この2人の社員の一方を報告先とする社員たちの立ち位置は変わらず、その社員たちの報告先のみが変化する。以下の例で社員4がチームリーダーに選ばれているとき、社員3と社員2を入れ替えることができるが、社員1と社員8を入れ替えることはできない。



新しいプロジェクトに従事することができる社員の人数の最大値と、実際にその人数の社員がプロジェクトに従事するために必要な入れ替え操作の最小回数を求めよ。

入力

入力の1行目は2つの整数 N, K からなり、それぞれ EGOI 社の社員数と社員が使うプログラミング言語の種類数を表す。

EGOI 社の社員には0から $N - 1$ まで番号がついており、CEO の Anneke の番号は0である。入力の2行目は N 個の整数 l_i ($0 \leq l_i < K$) からなり、社員 i が好むプログラミング言語を表す。

入力の続く $N - 1$ 行は社内の階層構造を表している。 i 行目は1つの整数 b_i ($0 \leq b_i < N$) からなり、社員 i の報告先の上司の番号を表す。ここで、 i は1から $N - 1$ までを動き、CEO の Anneke には上司がないことに注意する。

出力

2つの整数 P, S を1行に出力せよ。ここで、 P は入れ替え操作を繰り返して達成できる、新しいプロジェクトに従事する社員の人数（チームリーダーを含む）の最大値を表し、 S はこの最大値を達成するために必要な入れ替え操作の最小回数を表す。

制約・採点形式

- $1 \leq N \leq 10^5$.
- $1 \leq K \leq N$.

あなたの解答はいくつかの小課題においてテストされ、それぞれについて得点が定められている。それぞれの小課題はいくつかのテストケースを含む。ある小課題の得点を得るためには、その小課題に含まれるすべてのテストケースに対して正答する必要がある。

小課題	配点	制約
1	12	各 $1 \leq i < N$ に対し、社員 i の報告先は社員 $i - 1$ である
2	19	$K \leq 2$
3	27	各プログラミング言語に対し、その言語を好む社員の人数は高々 10 人である
4	23	$N \leq 2000$
5	19	追加の制約はない

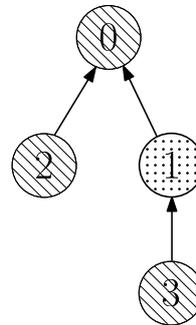
入出力例

最初の 2 つの入出力例では、社内の階層構造は次のようになっている。ここで、各頂点の様子はプログラミング言語を表している（斜線は言語 0，ドットは言語 1，白地は言語 2 を表す）。

Graph for example 1

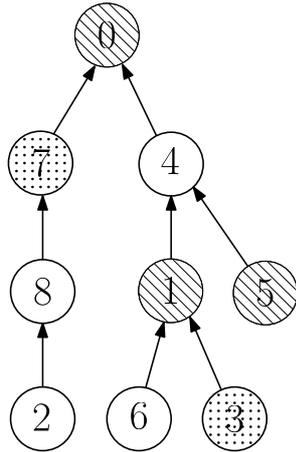


Graph for example 2

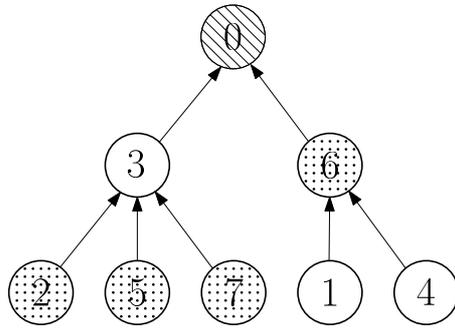


入出力例 1 では、社員 1 をチームリーダーに選び、同じプログラミング言語を好む社員 4 がプロジェクトに従事する。入れ替え操作を行ってより多くの人数を達成することはできない。入出力例 2 では、Anneke と同じプログラミング言語 0 を好む社員が 3 人いるため、Anneke をリーダーに選ぶことで、入れ替え操作を行わずに 3 人からなるチームを得られる。

Graph for example 3



Graph for example 4



入出力例3では、社員4をチームリーダーに選び、社員1と8、社員2と3を入れ替えることで、社員4と同じプログラミング言語2（白地）を好む4人の社員をチームに入れることができる。入出力例4では、社員6をチームリーダーに選び、社員4と7、社員1と5を入れ替えることで最大値を達成できる。ここで、最初にチームリーダーを固定する必要があるため、チームリーダーを選ぶ前に社員6と3を入れ替えて、スコア4を得ることはできないことに注意せよ。

標準入力	標準出力
<pre> 5 3 0 1 2 2 1 0 1 2 3 </pre>	<pre> 2 0 </pre>
<pre> 4 2 0 1 0 0 0 0 1 </pre>	<pre> 3 0 </pre>
<pre> 9 3 0 0 2 1 2 0 2 1 2 4 8 1 0 4 1 0 7 </pre>	<pre> 4 2 </pre>
<pre> 8 3 0 2 1 2 2 1 1 1 6 3 0 6 3 0 3 </pre>	<pre> 3 2 </pre>