

九月

杭州市の中央広場にはある有名な古木があり、この古木は N 頂点の根付き木と見なすことができる。頂点には 0 から $N - 1$ までの番号がつけられており、頂点 0 が根である。

子のない頂点は **葉** と呼ばれる。古木が葉を落とすとき、その時点で葉である頂点が 1 つ削除される。古木は 1 日に複数回葉を落とすかもしれない。

古木の管理を担当する M 人のボランティアがおり、 0 から $M - 1$ までの番号がつけられている。各ボランティアは独立に、以下の手順で古木が落とした葉を記録する：

毎日、その日に落ちたすべての葉の番号（すなわち、その日に削除されたすべての頂点の番号）を集め、その日より前の記録の後ろに任意の順で書き込む。

例えば、 1 日目に葉 $3, 4$ が落ちたとき、ボランティアは $3, 4$ または $4, 3$ を書き込む。その後、 2 日目に葉 $1, 2$ が落ちたとする。このとき、ボランティアは $1, 2$ または $2, 1$ を書き込み、最終的な記録は $(3, 4, 1, 2), (4, 3, 1, 2), (3, 4, 2, 1), (4, 3, 2, 1)$ のいずれかになる。

この手順は K 日間続き、**その間、毎日新しく葉が落ち**、最終的に古木には根である頂点のみが残った。

あなたは旅の途中、たまたま杭州を訪れることとなった。現在は寒い冬である。古木の葉の落ちた枝を見上げると、落ち葉が舞い散る美しい光景が想起される。

あなたはこの古木の落葉が何日間続いたかを知りたいが、 M 人のボランティアの記録しか見つけることができなかった。このとき、 K としてあり得る最大値を求めよ。

実装の詳細

あなたは以下の関数を実装する必要がある。

```
int solve(int N, int M, std::vector<int> F,  
          std::vector<std::vector<int>> S);
```

- N : 古木の頂点の数
- M : ボランティアの人数
- F : 整数からなる長さ N の配列。 $1 \leq i \leq N - 1$ を満たす整数 i について、 $F[i]$ は頂点 i の親の頂点の番号を表す。 $F[0]$ は常に -1 である。

- S : 長さ M の配列. S の各要素は, 整数からなる長さ $N - 1$ の配列である. $S[i][j]$ はボランティア i によって j 番目 (0 から始める) に記録された整数を表す.
- この関数は, 問題文のルールに従ったときの K としてあり得る最大値 (すなわち, 落葉のあった日数の最大値) を表す整数を返さなければならない.
- 1 つのテストケースの中で, 採点プログラムはこの関数を複数回呼ぶ可能性がある. それぞれの呼び出しは, 独立したシナリオとして処理されなければならない.

注意: この関数は複数回呼ばれる可能性があるため, 以前の呼び出しによって残されたデータ (特に, グローバル変数に保存されたデータ) の影響に注意せよ.

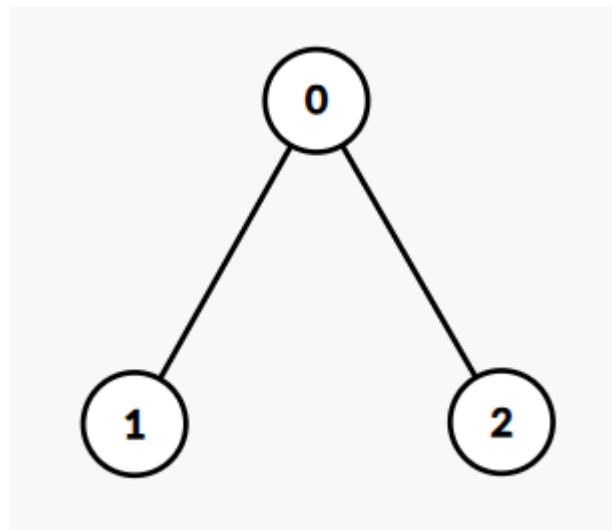
例

例 1

以下の呼び出しを考える:

```
solve(3, 1, {-1, 0, 0}, {{1, 2}});
```

対応する木は以下のように表される.



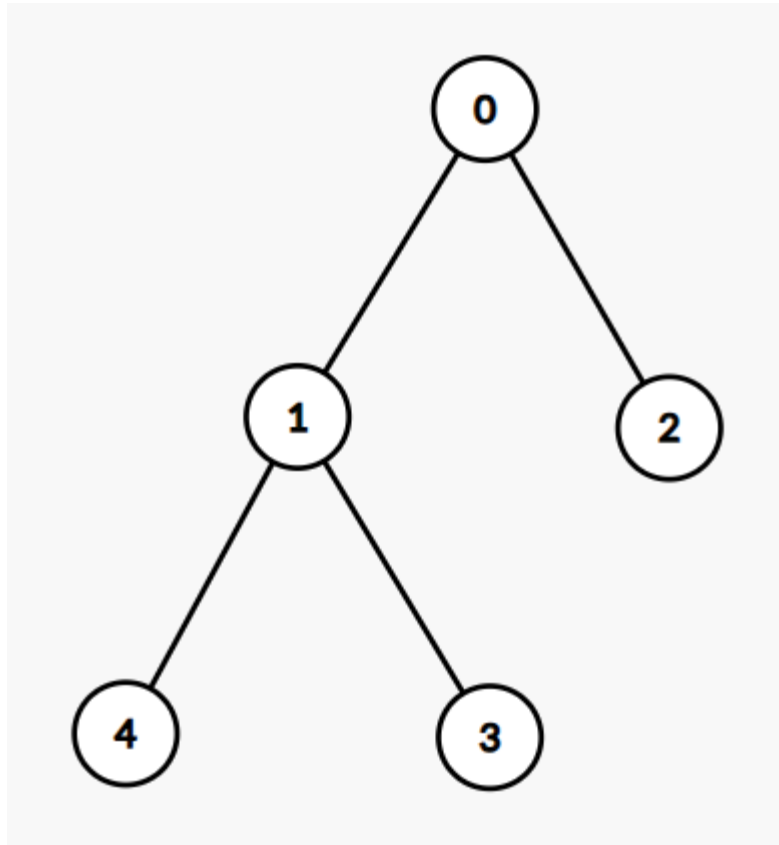
葉 1, 2 が同じ日に落ちた可能性もあるし, 1 日目に葉 1 が落ちた後, 2 日目に葉 2 が落ちた可能性もある. 落葉が 2 日間より多く続くことはあり得ないから, この関数は 2 を返さなければならない.

例 2

以下の呼び出しを考える:

```
solve(5, 2, {-1, 0, 0, 1, 1}, {{1, 2, 3, 4}, {4, 1, 2, 3}});
```

対応する木は以下のように表される.



落葉が 2 日以上続いたと仮定すると、2 人の記録から、葉 4 の落ちた日が複数ある (最初の日と最後の日) ことになり、矛盾する。

したがって、この関数は 1 を返さなければならない。

制約

- $2 \leq N \leq 10^5$.
- $1 \leq M \leq 5$.
- $\sum N M \leq 8 \times 10^5$.
- $F[0] = -1$. $1 \leq i \leq N - 1$ を満たす各 i について、 $0 \leq F[i] \leq i - 1$.
- $0 \leq i \leq M - 1$ を満たす各 i について、配列 $S[i]$ は $1, 2, \dots, N - 1$ の並べ替えである.
- F は頂点 0 を根とする根付き木を表す.

小課題

1. (11 点): $M = 1$, $N \leq 10$, $\sum N \leq 30$.
2. (14 点): $N \leq 10$, $\sum N \leq 30$.
3. (5 点): $M = 1$, $N \leq 1000$, $\sum N \leq 2000$, $F[i] = i - 1$.
4. (9 点): $M = 1$, $N \leq 1000$, $\sum N \leq 2000$.
5. (5 点): $N \leq 1000$, $\sum N \leq 2000$, $F[i] = i - 1$.
6. (11 点): $N \leq 1000$, $\sum N \leq 2000$.
7. (9 点): $M = 1$, $F[i] = i - 1$.
8. (11 点): $M = 1$.

9. (9 点): $F[i] = i - 1$.

10. (16 点): 追加の制約はない.

採点プログラムのサンプル

採点プログラムのサンプルは以下の形式で入力を読み込む：

- 1 行目: T

その後, T 個のテストケースが続く. 各テストケースでは, 以下の形式で入力を読み込む：

- 1 行目: $N M$
- 2 行目: $F[1] F[2] \dots F[N - 1]$
- $3 + i$ 行目 ($0 \leq i \leq M - 1$): $S[i][0] S[i][1] S[i][2] \dots S[i][N - 2]$

採点プログラムのサンプルは以下の形式であなたの答えを出力する：

各テストケースについて：

- 1 行目: solve の返り値