IOITC 2021

Tokens On A Tree

You are given a rooted tree with N nodes numbered $1, 2, \ldots, N$. Node 1 is the root node. Some of the nodes have a token in them. In one move, you can choose a non-root node that has a token, but it's parent doesn't, and shift the token from this node to its parent. What is the maximum number of moves you can make?

Input

- The first line contains T, the number of testcases. Each testcase contains three lines
- The first line of each testcase contains N, the number of nodes in the tree.
- The second line contains a string $S = s_1 s_2 \ldots s_N$. $s_i = 1$ if node has a token, 0 otherwise.
- The third line contains $N - 1$ space separated integers, $p_2, p_3, \ldots p_N$, where p_i is the parent of node i.

Output

For each testcase, print the maximum number of moves that can be made on a new line.

Test Data

In all inputs,

- $1 \leq N$
- The sum of N over all testcases doesn't exceed 10^6.
- For each valid i, $p_i < i$.

Subtask 1 (13 Points): $T \leq 10, N \leq 17$

Subtask 2 (18 Points): The sum of N over all testcases doesn't exceed 2000.

Subtask 3 (41 Points): The sum of N over all testcases doesn't exceed 10^5.

Subtask 4 (28 Points): No additional constraints

Sample Input

```
2
5
01010
1 1 3 3
5
10101
1 1 3 3
```

Sample Output

```
2
0
```

Explanation

In the first testcase, you can first move a token from node 4 to node 3, and then move a token from node 2 to node 1. In the second testcase, there are no possible valid moves.

Limits

Time: 2 seconds

Memory: 256 MB