Proofy and the cat

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

Proofy wanted to make his cat a fun game, so he drew with his pencil on a big piece of paper a tree of n vertices numbered from 1 to n that is rooted at vertex 1 (see notes for more clarification on the convention of rooted trees). He gave each edge a weight - which is a positive integer written on the edge. He also gave each vertex a value (he wrote it down), where the value of the ith vertex is a_i.

Now, he gave his cat a chip.

She can start playing *a round* by placing the chip on **any** vertex u.

In one round, she can play several moves (possibly none); that is, she stops the round whenever she likes.

On each move, she can move the chip from the vertex u along any edge (u, v) to another vertex v such that v is NOT the parent of u.

In the end of the round, she records all the weights of the edge she moved the chip along, and all the vertices that her chip visited.

Proofy tells her that the *profit of a round* is the **sum** of all values a_v for each **vertex** v her chip visited (including the starting one), and the *cost of a round* is the **maximum** of all **weights** of edges that her chip moved along (if the cat hasn't moved the chip along any edge, the cost is 0).

Now, the game is to find the **minimum cost** of a round such that its **profit** is **at least** k, for a positive integer k that Proofy gave his cat.

Can you help the cat find this minimum cost?

Important Note: if no round exists that has profit at least k, the cat should answer -1.

Input

The first line of input contains an integer t ($1 \le t \le 10^4$), the number of test cases.

The first line of each test case contains two integers n and k ($2 \le n \le 10^5, 1 \le k \le 10^9$), the number of vertices in the tree that Proofy drew and the lower-bound on the sum of a round that the cat has to collect.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$).

The third line of each test case contains $n-1$ integers $p_2, p_3, \ldots, p_n$ ($1 \le p_i < i$), where p_i is the parent of the ith vertex.

The fourth line of each test case contains $n-1$ integers $w_2, w_3, \ldots, w_n$ ($1 \le w_i \le 10^9$), where w_i is the weight of the edge from i to the parent of the ith vertex (i.e. the weight of the edge (p_i, i)).

The sum of n over all test cases doesn't exceed $5 \cdot 10^5$.

Output

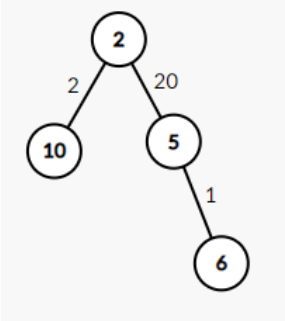
For each test case, output one line containing one integer, the minimum cost of a round such that its value is at least k, or -1 if such round doesn't exist.

Example

standard input	standard output
2 4 11 2 5 6 10 1 2 1 20 1 2 8 11 2 6 3 7 8 4 9 10 1 1 2 2 3 6 6 2 20 10 10 2 3 5	1 3

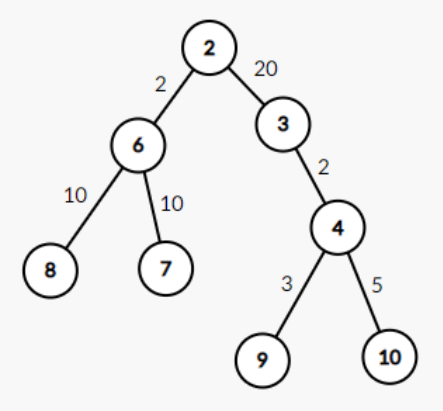
Note

In the first test case, the tree looks like this: (the labels are replaced by values for simplicity)



The optimal solution would be placing the chip on the node with value 5 and moving to the node with value 6, achieving a profit of a round of 11, which is at least $k = 11$, and the cost is 1.

In the second test case, the tree looks like this:



An optimal solution would be placing the chip on the node with value 3 and moving it to the node with value 4 and then to the node with value 9, and so the profit is 16, which is at least $k = 11$, and the cost is 3.

A *tree* is a connected undirected graph without cycles. A *rooted tree* is a tree with a selected vertex, which is called the *root*.

The tree is specified by an array of parents p containing n numbers: p_i is a parent of the vertex with the index i. The *parent* of a vertex u is a vertex that is the next vertex on the shortest path from u to the root.