Boleyn Salary

Boleyn Su runs a company called Acme. There are N employees in the company, and each one of them is represented by a unique employee id whose range lies in $[1, N]$. Being the head of company, Boleyn's employee id is 1.

Each employee, except Boleyn, has exactly one direct superior. This means that the hierarchial structure of the company is like a tree, where

1. Boleyn, employee id 1, represents the root node.
2. Each pair of employee is directly or indirectly connected to one another.
3. There is no cycle.

Let's represent the salary by the array $s = \{s[1], s[2], s[3]..., s[N]\}$, where $s[i]$ is the salary of the i^{th} employee. Salary structure in the company is non-uniform. Even a subordinate may get a higher salary than her superior. Some of the employees in Acme are curious about who gets the k^{th} lowest salary *among her subordinates*. Help them in solving their query.

Note

1. 1^{st} lowest salary is equivalent to lowest salary, 2^{nd} lowest means lowest salary which is greater that 1^{st} lowest salary, and so on.
2. Salary of each employee is different.
3. It is not necessary that the people who are placed higher on hierarchy will have a greater salary than their subordinates.

Input Format
The first line contains two space separated integers, N Q, where N is the number of employees in Acme, and Q is the number of queries.
Then follows $N-1$ lines. Each of these lines contain two space separated integers, u p, where p is the superior of u. u and p are employees id.
In the next line there are N space separated integers, $s[1]$ $s[2]$ ... $s[n]$, where $s[i]$, $i \in [1..N]$, is the salary of i^{th} employee.
Then, Q queries follow. Each query contains two space separated integers, v k. See output format for it's definition.

Output format
For the first query, print the id of employee who has the k^{th} lowest salary among the subordinates of v.
For the subsequent queries, we need to find the k^{th} lowest salary of the subordinates of $v+d$, where d is the answer of previous query.

Constraints
$1 \le N \le 3*10^4$
$1 \le Q \le 3*10^4$
$1 \le s[i] \le 10^9$, $i \in [1..N]$
$s[i] \neq s[j]$, $1 \le i < j \le N$

$1 \le u, p \le N, u \ne p$
$-N \le d \le N$
For 1^{st} query, $1 \le v \le N$
For later queries, $1 \le v+d \le N$
For each query, $1 \le K \le$ Number_of_subordinates

Sample Input

```
8 7
2 1
3 2
4 2
7 4
8 4
5 1
6 5
70 40 60 80 10 20 30 50
2 1
-6 5
-4 1
-5 3
2 1
-5 4
2 2
```

Sample Output

```
7
8
7
3
6
2
8
```

Explanation

Tree structure will be

```
         1(70)
        / \
       /   \
    2(40)  5(10)
     / \      \
    /   \      \
 3(60)  4(80)  6(20)
         / \
        /   \
     7(30)  8(50)
```

Query #1 `Node = 2`, `k = 1`: Subordinates, in increasing order of salary, are *(7, 30), (8, 50), (3, 60), (4, 80)*. So employee *7* has the *1*st lowest salary among the subordinates of *2*.
Query #2 `Node = -6+7 = 1`, `k = 5`: Subordinates are *(5, 10), (6, 20), (7, 30), (2, 40), (8, 50), (3, 60), (4, 80)* . *8*th employee has the *5*th lowest salary among the subordinate of *1*.
Query #3 `Node = -4+8 = 4`, `k = 1`: Subordinates are *(7, 30), (8, 50)* . Similarly 7 is the answer of this query.
Query #4 `Node = -5+7 = 2`, `k = 3`: Subordinates are *(7, 30), (8, 50), (3, 60), (4, 80)*. Similarly 3 is

the answer for this query.
Query #5 `Node = 2+3 = 5`, `k = 1`: Subordinates are *(6, 20)*. 6^{th} employee has the most, and only, lowest salary.
Query #6 `Node = -5+6 = 1`, `k = 4`: Subordinates are *(5, 10), (6, 20), (7, 30), (2, 40), (8, 50), (3, 60), (4, 80)*. 2 is answer of this query.
Query #7 `Node = 2+2 = 4`, `k = 2`: Subordinates are *(7, 30), (8, 50)*. Employee *8* has the second lowest salaries among the subordinates of 4.

Tested by: scturtle