**Recommendation Page**

Frankie wants to create a book-sharing website. People can write, share, and read many books. People have already posted **N** book numbered 1…**N**. Frankie needs help creating the recommendation feature of his website though and has asked for your help. Frankie would like to create a list of “suggested books” for every book.

In order to calculate the “relevance” between two books, He picks **N-1** pairs of books and manually computes their pairwise relevance. Frankie then visualizes the books as a network, where each book is a node and the **N-1** pairs of books he manually considered are connected. Frankie has also picked his **N-1** pairs so that any book can be reached from any other book along a path of connections in exactly one way. Frankie decides that the relevance of any pair of book should be defined as the minimum relevance of any connection along this path.

Frankie wants to pick a value **K** so that next to any book, all other books with relevance at least **K** to that book will be suggested. However, Frankie is worried that too many books will be suggested to his customers, which could distract them from other things! Therefore, he wants to carefully set an appropriate value of **K**. Frankie would like your help answering a number of questions about the suggested books for certain values of **K**.

**Input:** The first line of input contains **N** and **Q**. The next **N-1** lines each describe a pair of books that have been manually compared. Each line includes three integers *pi, qi,* and *ri*, indicating that books *pi* and *qi* are connected with relevance *ri*.

The next **Q** lines describe Frankie’s questions. Each line contains two integers, *ki* and *vi*, indicating that Frankie’s question asks how many books will be suggested to readers of book *vi* if **K** = *ki*.

**Output:** Output **Q** lines. Each line contains the answer to Frankie’s question for that test case.

**Example Input:**

4 3

1 2 3

2 3 2

2 4 4

1 2

4 1

3 1

**Example Output:**

3

0

2

**Explanation:** Frankie finds that books one and two have relevance three, that books two and three have relevance two, and that books two and four have relevance four. Based on this, books one and three have relevance min(3, 2) = 2, books one and four have relevance min(3, 4) = 3, and books three and four have relevance min(2, 4) = 2.

Frankie wants to know how many books will be suggested from book two if **K** = 1, from book one if **K** = 3, and from book one if **K** = 4. We see that with **K** = 1, books 1, 3, and 4 will be suggested on book two. With **K** = 4, no books will be suggested from book one. With **K** = 3, however, books 2 and 4 will be suggested from book one.