Lab – 8: Employee hierarchies

A company is providing a company database service, which maintains power hierarchies of all its employees. When employees login to the system, they are presented with a variety of services: searching, printing, querying, etc. One recent question that came up which the system was not quite prepared for was the following: “Which member of the company has joined most recently?” The employee who posed this question eventually had to answer it by manually searching the power tree database herself. The company decides to have a software written in case a similar question is asked in the future.

Input Format

The input starts with a number **N**. **N** lines follow, where each line is of the format <Name> <Tenure> <Salary> <Employee ID>. Each token is delimited by a space. Power relationship between two employees can be determined by the tuple (Tenure, Salary). A person with a higher tenure or a same tenure and a higher salary is considered to be more powerful.

The next line contains an integer **M**. **M** lines follow, where each line follows the format <Tenure> <Salary>. These are queries to find out the employee to which the tenure and salary corresponded to.

Output Format

Output M lines, where each line contains the employee id(s) of the employee(s) that matches the criteria. In case more than one employee matches the criteria, print all the employee ids separated by space. If no employee matches the criteria, print “NOT FOUND” instead.

Modeling the problem

Given that the company has a fairly large database consisting of tens of thousands of employees, it is important that the system responds quickly. It is also not possible to keep the records sorted by the power since the company keeps adding new employees at a fairly fast pace.

The code for representing the inputs as a binary search tree will be provided to you. Your main task will be answering the queries by using the binary search property of the tree.

Example

Input

5
asd 5 50000 12345
sdf 5 51000 34562
dfg 4 56000 23490
fgh 6 60000 90123
ghj 6 60000 67890
4
5 50000
5 51000
5 60000
6 60000

Output

12345
34562
NOT FOUND
90123 67890

Binary Search Tree representation of the sample inputs