

# Nested Binary Search Trees

locked

Problem	Submissions	Leaderboard	Discussions
---------	-------------	-------------	-------------

A binary search tree consists of nodes having the following data: a department key and a pointer to another binary search tree. The secondary binary search tree nodes contain ids of type integer. The nested binary search tree allows duplicate department keys, but does not allow duplicate pairs of department key and id. This means that there are no two elements containing the same department key and the same id. Given N pairs (department key and id) create the nested BST without balancing. You will then be given Q queries. For each query, print the number of nodes visited, and whether you found the query.

### Input Format

- The first line will contain the number of entries N and Q.
- The following N lines will contain two numbers each, department key and id to add.
- The following Q lines will contain two numbers, department key and id to search for.

### Constraints

- N is between 1 and 10^5
- Department key and ids are between 0 and 10^5
- Q is between 1 and 100

### Output Format

- Q lines each line containing the number of nodes visited and 1 if the entry is found or 0 if the entry is not found.

### Sample Input 0

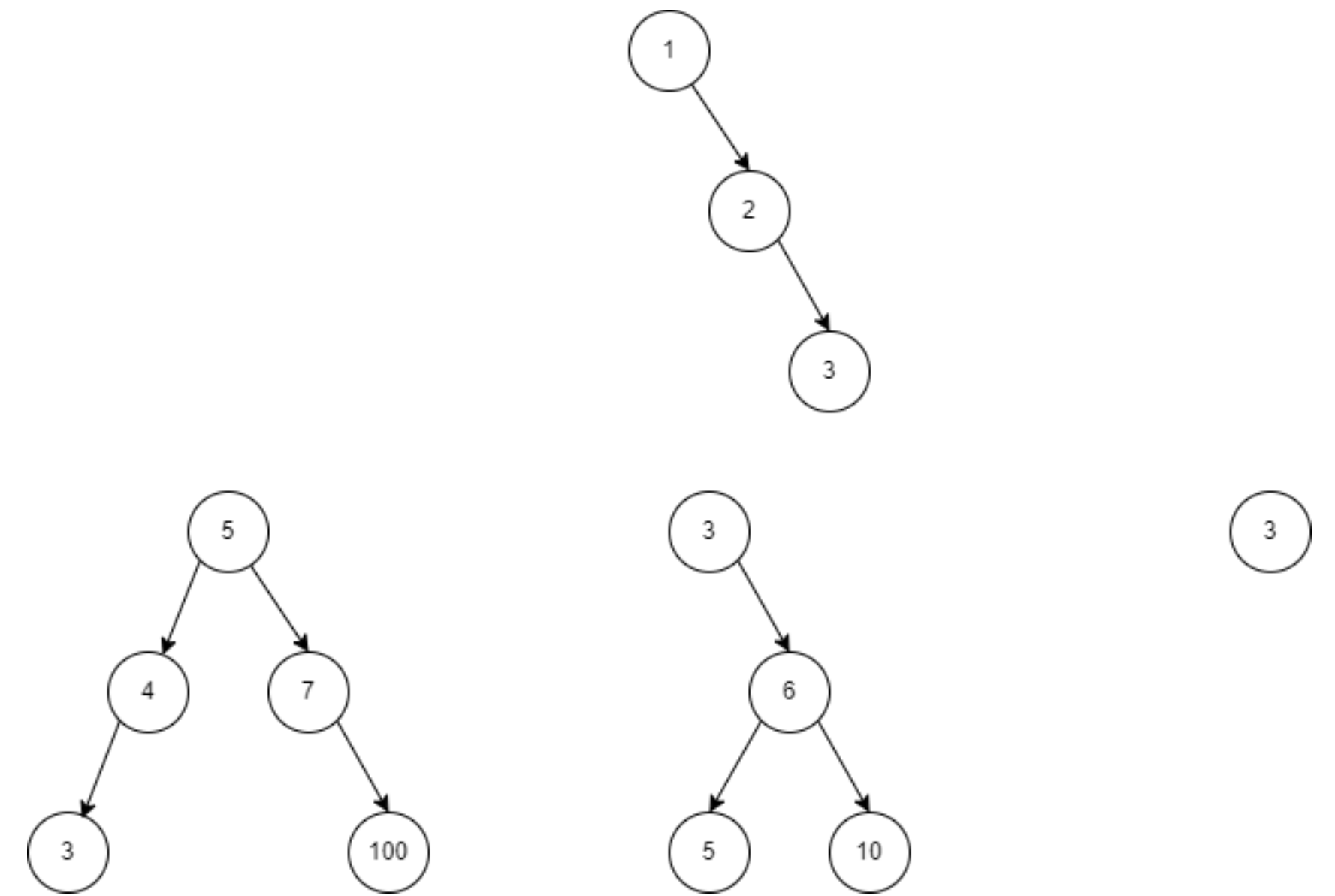
```
10 4
1 5
1 7
2 3
1 4
2 6
3 3
1 3
2 5
2 10
1 100
1 4
2 90
3 3
4 5
```

### Sample Output 0

```
3 1
5 0
4 1
3 0
```

### Explanation 0

You will create the following 4 trees, one main tree and one for each department key value. For each query trace the number of visited nodes and whether or not it was found.



C++14

```
1 #include <cmath>
2 #include <cstdio>
3 #include <vector>
4 #include <iostream>
5 #include <algorithm>
6 using namespace std;
7
8
9 int main() {
10     /* Enter your code here. Read input from STDIN. Print output to STDOUT */
11     return 0;
12 }
```

Line: 1 Col: 1