Binary Search Trees

1. Implement the Binary Search Tree for the following operations

   a. insert   b. get   c. min   d. max.   e. rank

Input Format:

- The first line of the input contains the number of **T** test cases.
- For each test case:
  - The first line of each test case contains n value. (Indicates the number of operations).
  - The operations are separated by spaces. (Check for the sample test case).
  - There will be a blank line for each test case.

Output Format:

- Print the output of the operation performed on the BST.
  - For insert operation, print the BST nodes in inorder traversal. Each Node separated by comma and the key and value are separated by colon (:)
  - For get operation, print the value returned by get method.
  - For min operation, print the min node in the BST, print key and value separated by colon, otherwise, raise exception with the message called min()with empty symbol table.
  - For max operation, print the max node in the BST, print key and value separated by colon, otherwise, raise exception with the message called max()with empty symbol table.
  - For rank operation, print the rank of the Key.
- Note: Key and value are separated with colon (:) and pair is separated with comma.

Constraints:

- 1 ≤ T ≤ 10. (Test Cases)

Sample Input:

```
1
18
insert dawn 90
insert mike 8
min
get dave
insert dave 6
get dave
max
min
insert beth 0
insert cind 5
min
insert gina 99
get cind
insert pat 77
max
insert sue 100
```

```
max
rank Robert
```

Sample Output:

```
dawn:90
dawn:90,mike:8
dawn:90
no such key
dave:6,dawn:90,mike:8
6
mike:8
dave:6
beth:0,dave:6,dawn:90,mike:8
beth:0,cind:5,dave:6,dawn:90,mike:8
beth:0
beth:0,cind:5,dave:6,dawn:90,gina:99,mike:8
5
beth:0,cind:5,dave:6,dawn:90,gina:99,mike:8,pat:77
pat:77
beth:0,cind:5,dave:6,dawn:90,gina:99,mike:8,pat:77,sue:100
sue:100
7
```