Below is the written test for Bongo’s Python Engineer position. Please read through the entire test before starting to write it.

1) Write the following function’s body. A nested dictionary is passed as parameter. You need to print all keys with their depth.

Sample Input:

```
a = {
        “key1”: 1,
        “”key2”: {
                “key3”: 1,
                “key4”: {
                        “key5”: 4
                }
        }
}
```

Sample Output:

```
key1 1
key2 1
key3 2
key4 2
key5 3
```

```
def print_depth(data):
        # Write function body
```

You may write additional function.

2) Write a new function with same functionality from Question 1, but it should be able to handle a Python object in addition to a dictionary from Question 1.

Sample Input:

```
class Person(object):
    def __init__(self, first_name, last_name, father):
        self.first_name = first_name
        self.last_name = last_name
        self.father = father

person_a = Person(“User”, “1”, none)
person_b = Person(“User”, “2”, person_a)

a = {
    “key1”: 1,
    ”key2”: {
        “key3”: 1,
        “key4”: {
            “key5”: 4,
            “user”: person_b,
        }
    },
}
```

Sample Output:

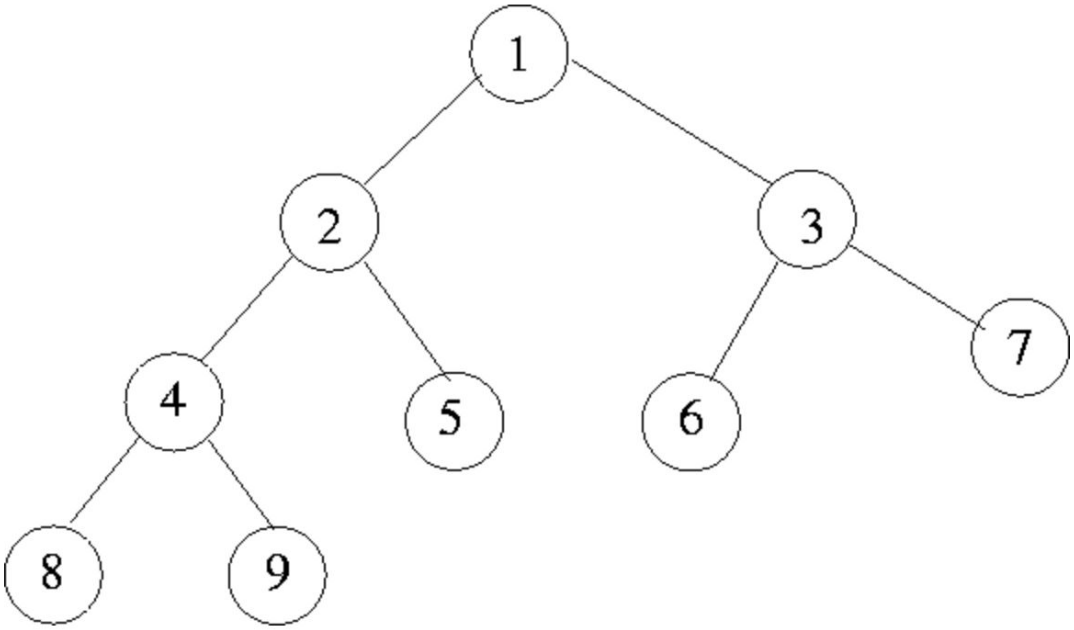
```
key1 1
key2 1
key3 2
key4 2
key5 3
user: 3
first_name: 4
last_name: 4
father: 4
first_name: 5
last_name: 5
father: 5
```

```
def print_depth(data):
    # Write function body
```

You may write additional function.

3) Write following functions body. 2 Nodes are passed as parameter. You need to find Least Common Ancestor and print its value. Node structure are as following:

```
class Node{
        value;
        parent;
}
```



Ancestor Definition:

1) Any node falls under parent chain till root node.
2) A node is an ancestor of itself.

For example: if we consider Node 7 it’s ancestors will be 1, 3, and 7.

All nodes values are unique for this tree.

You function needs to find least common ancestor (closest common ancestor). For an example for the tree image,
if 6 and 7 passed to lca it should return 3
if 3 and 7 passed to lca it should return 3

```
def lca(node1, node2):
        # Write function body
```

You may write additional function.
Explain the Runtime and Memory requirements for your solution.

Submission:

1) Implement solution for these problems.
2) You are encouraged to write unit test for all three problems.
3) Upload to github/bitbucket or any other code sharing platform.
4) Send an email to hr@bongobd.com with subject “Bongo Python Code Test” with your code repository URL in the email body.

If you have any questions, please send them to mirza.asif@bongobd.com with a subject line of “Questions on Bongo Code Test”.