problem_4_active_directory

March 11, 2020

0.0.1 Analyze:

I need to find a recursion solution with a cache to solve the problem. recoding the check result by dict. if the group already be checked. didn't check the group again.

Subtask: - Design recursion solution to check all groups - Define a cache to store the check result

The function **is_user_in_group_recursion** takes O(n) time complexity, I think the "for loop" is O(n) and "recursion solution"is O(log(n)). using cache to check, the time complexity isn't O(nlog(n))

I think it's linear space complexity O(n) , 2 list(),1 dict().

[ ]:
```
class Group(object):
    def __init__(self, _name):
        self.name = _name
        self.groups = []
        self.users = []

    def add_group(self, group):
        self.groups.append(group)

    def add_user(self, user):
        self.users.append(user)

    def get_groups(self):
        return self.groups

    def get_users(self):
        return self.users

    def get_name(self):
        return self.name
```

[161]:
```
def is_user_in_group(user, group):
    """
    Return True if user is in the group, False otherwise.

    Args:
      user(str): user name/id
```

```
        group(class:Group): group to check user membership against
    """
    groups_dict = dict()
    result = is_user_in_group_recursion(user, group, groups_dict)
    groups_dict[group.get_name()] = result
    return True in groups_dict.values()
```

[1]:
```
def is_user_in_group_recursion(user, group, groups_dict):
    if user in group.get_users():
        return True
    else:
        for sub_group in group.get_groups():
            if sub_group.get_name() not in groups_dict:
                result = is_user_in_group_recursion(user, sub_group,␣
↪groups_dict)
                groups_dict[sub_group.get_name()] = result
    return False
```

[159]:
```
parent = Group("parent")
child = Group("child")
sub_child = Group("subchild")

sub_child_user = "sub_child_user"
sub_child.add_user(sub_child_user)

child.add_group(sub_child)
parent.add_group(child)


print('The sub_child_user in parent is {}'.
↪format(is_user_in_group("sub_child_user", parent)))
print('The sub_child_user in child is {}'.
↪format(is_user_in_group("sub_child_user", child)))
print('The sub_child_user in sub_child is {}'.
↪format(is_user_in_group("sub_child_user", sub_child)))
```

```
The sub_child_user in parent is True
The sub_child_user in child is True
The sub_child_user in sub_child is True
```

[160]:
```
parent = Group("parent")

child_1 = Group("child1")
child_2 = Group("child2")

sub_child_1 = Group("subchild1")
sub_child_2 = Group("subchild2")
```

```
sub_child_3 = Group("subchild3")

sub_sub_child_1 = Group("subsubchild1")
sub_sub_child_3 = Group("subsubchild3")
sub_sub_child_3.add_user("sub_sub_child_3_user")

sub_child_1.add_group(sub_sub_child_1)
sub_child_3.add_group(sub_sub_child_3)

child_1.add_group(sub_child_1)
child_2.add_group(sub_child_2)
child_2.add_group(sub_child_3)

parent.add_group(child_1)
parent.add_group(child_2)

# should return True , sub_sub_child_3_user in sub_sub_child_3.users[]
print('The sub_sub_child_3_user in sub_sub_child_3 is {}'.
 ↪format(is_user_in_group("sub_sub_child_3_user", sub_sub_child_3)))

# should return True , sub_sub_child_3 's parent is sub_child_3
print('The sub_sub_child_3_user in sub_child_3 is {}'.
 ↪format(is_user_in_group("sub_sub_child_3_user", sub_child_3)))

# should return True , sub_sub_child_3 's grandparents is child_2
print('The sub_sub_child_3_user in child_2 is {}'.
 ↪format(is_user_in_group("sub_sub_child_3_user", child_2)))

# should return True , the parent is root group
print('The sub_sub_child_3_user in parent is {}'.
 ↪format(is_user_in_group("sub_sub_child_3_user", parent)))

# should return False, the sub_sub_child_3 's parent is sub_child_3
print('The sub_sub_child_3_user in sub_child_1 is {}'.
 ↪format(is_user_in_group("sub_sub_child_3_user", sub_child_1)))

# should return False, the sub_sub_child_3 's parent is sub_child_3
print('The sub_sub_child_3_user in sub_child_2 is {}'.
 ↪format(is_user_in_group("sub_sub_child_3_user", sub_child_2)))

# should return False, the sub_sub_child_3 's parent is sub_child_3
print('The sub_sub_child_3_user in child_1 is {}'.
 ↪format(is_user_in_group("sub_sub_child_3_user", child_1)))
```

```
The sub_sub_child_3_user in sub_sub_child_3 is True
The sub_sub_child_3_user in sub_child_3 is True
The sub_sub_child_3_user in child_2 is True
```

```
The sub_sub_child_3_user in parent is True
The sub_sub_child_3_user in sub_child_1 is False
The sub_sub_child_3_user in sub_child_2 is False
The sub_sub_child_3_user in child_1 is False
```