**Active Directory**

In Windows Active Directory, a group can consist of user(s) and group(s) themselves. We can construct this hierarchy as such. Where User is represented by str representing their ids.

**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

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)

Write a function that provides an efficient look up of whether the user is in a group.

**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

"""

**return** **None**