Algorithm for file updates in Python

Project description

to create an algorithm that uses Python code to check whether the allow list contains any IPs identified on the remove list.

Open the file that contains the allow list

```
4
5  with open("allow_list.txt", "r") as file:
6      import_file = file.read()
7
```

Read the file contents

```
4
5  with open("allow_list.txt", "r") as file:
6      import_file = file.read()
7
```

Convert the string into a list

```
5  with open("allow_list.txt", "r") as file:
6      import_file = file.read()
7
8  allowed_users = import_file.split()
9
```

Iterate through the remove list

```
with open("allow_list.txt", "r") as file:
    import_file = file.read()

allowed_users = import_file.split()

for user in allowed_users:
    if user in remove_list:
        allowed_users.remove(user)

```

Remove IP addresses that are on the remove list

```
with open("allow_list.txt", "r") as file:
    import_file = file.read()

allowed_users = import_file.split()

for user in allowed_users:
    if user in remove_list:
        allowed_users.remove(user)

```

Update the file with the revised list of IP addresses

```
with open("allow_list.txt", "r") as file:
    import_file = file.read()

allowed_users = import_file.split()

for user in allowed_users:
    if user in remove_list:
        allowed_users.remove(user)

updated_allowed_users = " ".join(allowed_users)

with open("allow_list.txt", "w") as file:
    file.write(updated_allowed_users)
```

Summary

```
with open("allow_list.txt", "r") as file:
    import_file = file.read()

allowed_users = import_file.split()

for user in allowed_users:
    if user in remove_list:
        allowed_users.remove(user)

updated_allowed_users = " ".join(allowed_users)

with open("allow_list.txt", "w") as file:
    file.write(updated_allowed_users)
```

- This script first imports an allow list into Python, and the variable import_file is assigned to the allow_list.
- The imported file is then assigned to the variable, allowed_users, and is then turned into a list.
- An iterative statement then loops over the users, looking for any that are also in the removed_users variables (which must be added to make it work).
- Then this new and updated list is turned back into a string.
- Finally, the allow_list file is updated with the new allowed users, effectively removing all of the users that needed to be removed.