Algorithm for file updates in Python

Project description

I developed an algorithm to manage access control for a content system. The "allow_list.txt" file contains authorized IP addresses, while the "remove list" contains IPs that should no longer have access. The algorithm automatically updates the "allow_list.txt" by removing the unauthorized IPs, ensuring only permitted addresses can access the content.

Open the file that contains the allow list

```
In [13]: # Assign `import_file` to the name of the file
         import_file = "allow_list.txt"
         # Assign `remove_list` to a list of IP addresses that are no longer allowed to access restricted information.
         remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]
         # First line of `with` statement
         with open(import_file, "r") as file:
```

Using with open(import_file, “r”) as file: , I set the import file as a readable file.

Read the file contents

```
In [3]: # Assign `import_file` to the name of the file
        import_file = "allow_list.txt"
        # Assign `remove_list` to a list of IP addresses that are no longer allowed to access restricted information.
        remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]
        # Build `with` statement to read in the initial contents of the file
        with open(import_file, "r") as file:
          # Use `.read()` to read the imported file and store it in a variable named `ip_addresses`
          ip_addresses = file.read()
        # Display `ip_addresses`
        print(ip_addresses)
        ip
        address
        192.168.25.60
        192.168.205.12
```

In this one I utilized .read(). Ip_addresses = file.read() to read the file of ip addresses.

Convert the string into a list

```
In [3]: # Assign `import_file` to the name of the file

        import_file = "allow_list.txt"

        # Assign `remove_list` to a list of IP addresses that are no longer allowed to access restricted information.

        remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]

        # Build `with` statement to read in the initial contents of the file

        with open(import_file, "r") as file:

          # Use `.read()` to read the imported file and store it in a variable named `ip_addresses`

          ip_addresses = file.read()

        # Display `ip_addresses`

        print(ip_addresses)
```

```
ip
address
192.168.25.60
192.168.205.12
192.168.6.9
192.168.52.90
192.168.90.124
192.168.186.176
```

Here I turned the data from string to list type. Using .split().

Iterate through the remove list

```
# Use `.split()` to convert `ip_addresses` from a string to a list

ip_addresses = ip_addresses.split()

# Build iterative statement
# Name loop variable `element`
# Loop through `remove_list`

for element in remove_list:

  # Display `element` in every iteration

  print(element)
```

```
192.168.97.225
192.168.158.170
192.168.201.40
192.168.58.57
```

Here I had to remove the elements of remove_list from the ip_addresses. It required using an iterative statement.

Remove IP addresses that are on the remove list

```
for element in remove_list:

    # Create conditional statement to evaluate if `element` is in `ip_addresses`

    if element in ip_addresses:

        # use the `.remove()` method to remove
        # elements from `ip_addresses`

        ip_addresses.remove(element)

# Display `ip_addresses`

print(ip_addresses)
```

```
['ip', 'address', '192.168.25.60', '192.168.205.12', '192.168.6.9', '192.168.52.90', '192.168.90.124', '192.168.186.176', '192.
168.133.188', '192.168.203.198', '192.168.218.219', '192.168.52.37', '192.168.156.224', '192.168.60.153', '192.168.69.116']
```

Removed the ip addresses that were part of the remove_list.

Update the file with the revised list of IP addresses

```
for element in remove_list:

    # Create conditional statement to evaluate if `element` is in `ip_addresses`

    if element in ip_addresses:

        # use the `.remove()` method to remove
        # elements from `ip_addresses`

        ip_addresses.remove(element)

# Convert `ip_addresses` back to a string so that it can be written into the text file

ip_addresses = "\n".join(ip_addresses)

# Build `with` statement to rewrite the original file

with open(import_file, "w") as file:

  # Rewrite the file, replacing its contents with `ip_addresses`

  file.write(ip_addresses)
```

Summary

I built an algorithm to open and read a file. After that I deleted the ip addresses that were on the remove list from the file.