Algorithm for file updates in Python

Working with Files in Python

Here, we are updating files for employees who can access restricted content. There are different lists for who can and can't work with specific files. The goal is to grant and remove access from the specific employees based on their IP addresses.

Open the file that contains the allow list

```
import_file = "allow_list.txt"
remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]
with open(import_file, "r") as file:
```

- There is an allow list file that is used to contain a list of all IP addresses for employees who have authorized access. That allow list file will be stored in a variable named "import_file".
- The remove_list variable contains a list of the IP addresses that must have access removed from the allowed list.
- The open file is not fully completed yet in the code above.

Read the file contents

```
# 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)
```

- Now, with the displayed code, we have opened the import file (allow_list.txt) and have elected to read the contents.
- Using the .read() method to read the imported file, we are storing this in the variable called "ip_addresses".

- When you print out the results of ip_addresses, it will display the whole list of the IP addresses in the form of a string.
- The import_file IP addresses currently have both allowed and those that need to be removed.

Convert the string into a list

```
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()

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

ip_addresses = ip_addresses.split()

# Display `ip_addresses`

print(ip_addresses)
```

```
['ip_address', '192.168.25.60', '192.168.205.12', '192.168.97.225', '192.168.6.9', '192.168.52.96
8.90.124', '192.168.186.176', '192.168.133.188', '192.168.203.198', '192.168.201.40', '192.168.21
168.156.224', '192.168.60.153', '192.168.58.57', '192.168.69.116']
```

- The goal now is to convert the singular string of IP addresses that was returned into a list of strings. The .split() method is used for this.
- The string is turned into a list so we may iterate through the list of strings to remove the necessary IP addresses.
- ip_addresses.split() will first read the file contents again, then will split the singular string into multiple strings of different IP addresses.
- The output at the bottom then displays that all IP addresses in the import_file have been converted into multiple strings.

Iterate through the remove list

```
for element in ip_addresses:
```

In order to remove the elements from the ip_addresses, we need to create an iterative statement.

- The above code only displays the header.
- The for loop iterates through the ip_addresses. The loop variable is element
- We are doing this because within this function, we will be cross referencing the matched IP addresses in the import_file and remove_list.

Remove IP addresses that are on the remove list

```
for element in ip_addresses:

  # Build conditional statement
  # If current element is in `remove_list`,

    if element in remove_list:

        # then current element should be removed from `ip_addresses`

        ip_addresses.remove(element)

# Display `ip_addresses`

print(ip_addresses)
```

- Within the body of the for iterative statement, we have a conditional statement. This conditional statement is so that it can iterate through the remove_list, in conjunction with the ip_addresses to find matches to be removed.
- The conditional if statement is searching for elements in the remove_list.
- If an element in the remove_list is within the ip_addresses list, that address will then be removed. This is done because the .remove() method will remove the targeted element that is passed into it. The value being passed into the .remove() method in this instance is element.
- Ex: ip_addresses.remove(element)
- This is possible because there are no duplicates in the list of IP addresses.

Update the file with the revised list of IP addresses

```
ip_addresses = " ".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)
```

Now we need to update the original file with the revised list. This revised list will not include the removed IP's since that was completed in the prior step.

- First, the string needs to convert back to a singular string so the contents of the file are saved properly. This is done by using the .join() method.
- ip_addresses = " ".join(ip_addresses) will convert all remaining IP addresses into a single string by joining them together.

- We then open the file again but instead of reading the file, we are updating it so we pass in the “w” parameter. This allows us to write, revise, or update a file.
- Then we use file.write(ip_addresses) to rewrite the file with the update contents of ip_addresses.

Summary

```
import_file = "allow_list.txt"
remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]
with open() as file:
ip_addresses = ip_addresses.split()
```

Above are some of the main components of this project. We needed to know which files to access and how to revise/update them. The target IP’s to remove were done by utilizing different methods and functions. Iterative and conditional statements allow us to make our code more complex but concise, so we may access loads of data effectively. By completing this project, we have now made all appropriate changes within the files needed.