Algorithm for file updates in Python

Project Description

My organization wants me to create an algorithm to automate updating a file "allow_list.txt" by removing IP addresses that should no longer have access to the restricted content in this file.

Open the File that Contains the Allow List

The first step is to open the text file containing the IP addresses that are allowed to access restricted information. I used the `open()` function with the "r" parameter to open the file in read mode.

```
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:
```

Read the File Contents

After opening the file, I used the `.read()` method to read the entire contents of the file into a variable called `ip_addresses`. This allows me to work with the data stored in the file.

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

Convert the String into a List

The `.read()` method returns the file contents as a single string. To work with individual IP addresses, I used the `.split()` method to convert the string into a list, where each IP address is a separate element.

```
ip_addresses = ip_addresses.split()
```

Iterate Through the Remove List

To identify and remove IP addresses that should no longer have access, I created a loop that iterates through each element in the `ip_addresses` list.

```
for element in ip_addresses:
```

Remove IP Addresses that are on the Remove List

Inside the loop, I added a conditional statement to check if the current element is in the `remove_list`. If it is, I used the `.remove()` method to remove that IP address from the `ip_addresses` list.

```
for element in ip_addresses:
  if element in remove_list:
    ip_addresses.remove(element)
```

Update the File with the Revised List of IP Addresses

After removing the unauthorized IP addresses, I needed to update the original file. First, I converted the `ip_addresses` list back to a string using the `.join()` method. Then, I opened the file in write mode ("w") and used the `.write()` method to overwrite the file with the updated list.

```
ip_addresses = " ".join(ip_addresses)

with open(import_file, "w") as file:
  file.write(ip_addresses)
```

Summary

I created a function called `update_file()` that combines all these steps to automate the process of removing unauthorized IP addresses from the allow list file. The function takes two parameters: the name of the file to update and a list of IP addresses to remove.

```
def update_file(import_file, remove_list):
    with open(import_file, "r") as file:
        ip_addresses = file.read()

    ip_addresses = ip_addresses.split()

    for element in ip_addresses:
        if element in remove_list:
            ip_addresses.remove(element)

    ip_addresses = " ".join(ip_addresses)

    with open(import_file, "w") as file:
        file.write(ip_addresses)
```

This function can be called with different file names and different lists of IP addresses to remove, making it a flexible and reusable solution for maintaining access control lists. By automating this process, I've helped improve the security of our organization's restricted content while saving time and reducing the risk of human error.

THE FULL JUPYTER PYTHON DOCUMENT IS ATTACHED BELOW