Portfolio: Automating IP Address Updates with Python Algorithm

Project Description

In the context of managing access to restricted content at my organization through an IP allow list, I developed a Python algorithm to streamline the process of updating the "allow_list.txt" file. This algorithm automatically removes IP addresses from the list based on a predefined remove list, ensuring accurate and efficient content access control.

Algorithm Workflow

Step 1: Opening the Allow List File

```
import_file = "allow_list.txt"

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

I began by opening the "allow_list.txt" file in read mode and converting its content into a list of IP addresses using the `.read()` method and `.split()` function.

Step 2: Iterating Through the Remove List

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

I used a `for` loop to iterate through the `remove_list` of IP addresses to be removed. Within the loop, I checked if each IP address was present in the `ip_addresses` list and removed it using the `.remove()` method.

Step 3: Converting List Back to String

```
updated_ip_addresses = "\n".join(ip_addresses)
```

After updating the `ip_addresses` list, I used the `.join()` method to convert it back into a string format, with each IP address separated by a newline character.

Step 4: Updating the Allow List File

```
with open(import_file, 'w') as file:
    file.write(updated_ip_addresses)
```

Finally, I opened the "allow_list.txt" file in write mode and updated its contents with the revised list of IP addresses using the `.write()` method.

Summary

I designed an algorithm that automates the process of updating the "allow_list.txt" file, which controls access to restricted content through IP addresses. This algorithm efficiently removes specified IP addresses from the list while maintaining accuracy. By leveraging file handling, list manipulation, and conditional statements, I created an effective solution that enhances access control management in real-world scenarios.

Outcomes

- Efficiency: The algorithm significantly reduces the manual effort required for updating the IP allow list.
- Accuracy: By automating the removal of IP addresses, the algorithm ensures precise and error-free updates.
- Scalability: The solution can effortlessly handle an increasing number of IP addresses in the allow list.

Reflection

This project allowed me to showcase my proficiency in Python programming, file handling, list operations, and conditional statements. The algorithm's successful implementation demonstrates my ability to solve practical challenges using code, contributing to improved efficiency and accuracy in access control procedures.