Handling Files in python

Project description

In this task we design an algorithm, taht address a common task for a security analyst at a healthcare company. The task involves checking the "allow_list.txt" file, which contains IP addresses of employees with access to restricted content, and removing IP addresses listed in the "remove_list." The algorithm ensures that unauthorized IP addresses are removed, maintaining the integrity of the access control list.

Open the file that contains the allow list

```
import_file = "allow_list.txt"
with open(import_file, "r") as file:
  # Work with the file inside this block
```

Read the file contents

```
  ip_addresses = file.read()
```

Convert the string into a list

```
  ip_addresses = ip_addresses.split("\n")
```

Iterate through the remove list

```
  remove_list = ["192.168.1.100", "10.0.0.5", "172.16.0.2"]  # Example remove_list
  for element in remove_list:
```

Remove IP addresses that are on the remove list

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

It's important to note that the remove method is used to remove IP addresses because there are no duplicates in the ip_addresses list.

Update the file with the revised list of IP addresses

```
updated_content = "\n".join(ip_addresses)
with open(import_file, "w") as file:
    file.write(updated_content)
```

Summary

This algorithm checks the "allow_list.txt" file for unauthorized IP addresses found in the "remove_list" and updates the file by removing these addresses. It begins by opening the file, reading its contents, and converting them into a list. Then, it iterates through the "remove_list" and removes any matching IP addresses from the list. Finally, the updated list is converted back to a string and written back to the file.

By implementing this algorithm, we can efficiently manage access control lists, ensuring that only authorized employees have access to restricted content, which is crucial for maintaining data security and privacy in a healthcare environment.