# Algorithm for file updates in Python

## Project description

In this algorithm, we 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.