

# Algorithm for file updates in Python

## Project Overview:

At my organization, access to restricted content is controlled through an IP allow list maintained in the "allow\_list.txt" file. A separate remove list identifies IPs that should no longer have access. I developed a Python script to automate removing these IPs from the allow list.

## Open and Read the Allow List:

First, the script opens "allow\_list.txt" in read mode using a with statement to manage resources efficiently and ensure the file closes automatically.

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

The .read() method converts the file contents into a string, which is then stored in ip\_addresses.

## Convert the String to a List:

To easily manipulate individual IPs, the string is split into a list of IP addresses.

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

This splits the string by whitespace, converting it into a list format for easy removal of IPs.

## Iterate Over the Remove List:

The script then loops through the IPs in remove\_list, checking if each IP exists in ip\_addresses. If found, it removes the IP using the .remove() method.

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

## Update the Allow List:

After processing the removals, the list is converted back to a string with each IP on a new line, using the `.join()` method. The script then writes the updated list back to "allow\_list.txt".

```
with open("allow_list.txt", "w") as file:  
    file.write("\n".join(ip_addresses))
```

## Summary:

This Python algorithm automates the removal of IPs from the allow list by reading the file, converting its contents into a list, removing IPs as specified in the remove list, and updating the file with the revised IPs.