Algorithm for file updates in Python

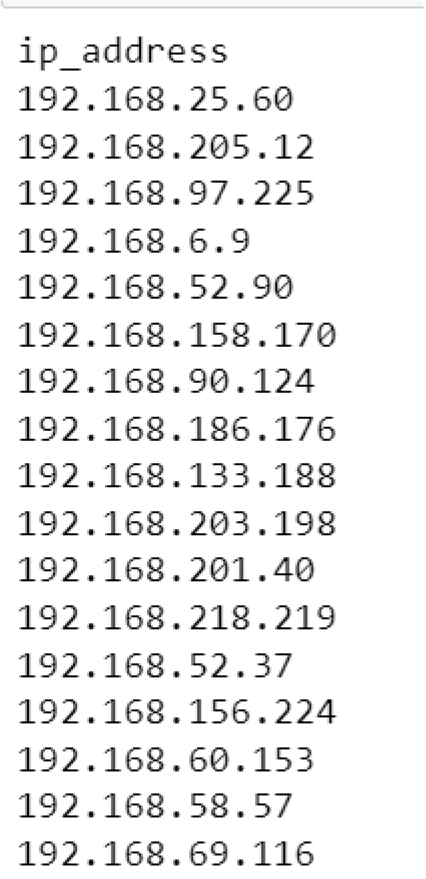
Project description

As a security professional at a healthcare company, I'm tasked with maintaining an "allow list" of IP addresses that can access sensitive patient data. This list needs to be regularly updated based on employee roles and access requirements. My job is to create a Python script that automates this process.

Open the file that contains the allow list

```
# Assign `import_file` to the name of the file

import_file = "allow_list.txt"

# Assign `remove_list` to a list of IP addresses that are no longer allowed to access restricted information.

remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]

# Build `with` statement to read in the initial contents of the file

with open(import_file, "r") as file: # Use `open()` with the "r" parameter

  # Use `.read()` to read the imported file and store it in a variable named `ip_addresses`

  ip_addresses = file.read() # Assign the output of `.read()` to `ip_addresses`

# Display `ip_addresses`

print(ip_addresses) # Display the contents of `ip_addresses`
```

Read the file contents



```
ip_address
192.168.25.60
192.168.205.12
192.168.97.225
192.168.6.9
192.168.52.90
192.168.158.170
192.168.90.124
192.168.186.176
192.168.133.188
192.168.203.198
192.168.201.40
192.168.218.219
192.168.52.37
192.168.156.224
192.168.60.153
192.168.58.57
192.168.69.116
```

Convert the string into a list

```
# Use `.split()` to convert `ip_addresses` from a string to a list

ip_addresses = ip_addresses.split()

# Display `ip_addresses`

print(ip_addresses)
```

```
['ip_address', '192.168.25.60', '192.168.205.12', '192.168.97.225',
'192.168.6.9', '192.168.52.90', '192.168.158.170', '192.168.90.124',
'192.168.186.176', '192.168.133.188', '192.168.203.198', '192.168.201.40',
'192.168.218.219', '192.168.52.37', '192.168.156.224', '192.168.60.153',
'192.168.58.57', '192.168.69.116']
```

Iterate through the remove list

```
remove_list = ["192.168.1.3", "10.0.0.200"]

for element in remove_list:
  # Code to process each element in remove_list will go here
```

Remove IP addresses that are on the remove list

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

```
['ip_address', '192.168.25.60', '192.168.205.12', '192.168.6.9',
'192.168.52.90', '192.168.90.124', '192.168.186.176', '192.168.133.188',
'192.168.203.198', '192.168.218.219', '192.168.52.37', '192.168.156.224',
'192.168.60.153', '192.168.69.116']
```

Update the file with the revised list of IP addresses

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

```
ip_address 192.168.205.12 192.168.6.9 192.168.52.90 192.168.90.124 192.168.186.176 192.168.133.188
192.168.218.219 192.168.52.37 192.168.156.224 192.168.60.153 192.168.69.116
```

Summary

This Python algorithm efficiently updates an allow list of IP addresses by removing unauthorized entries. It leverages file handling, string manipulation, list operations, and conditional logic to achieve this. The `with` statement ensures safe file access, while the `.read()`, `.split()`, `.remove()`, `.join()`, and `.write()` methods enable seamless interaction with the file's contents. By automating this process, the script helps maintain data security and access control within the healthcare company.