Algorithm for file updates in Python

Ulysses Danforth Bakolias

Project Outline:
Working as a security specialist in the healthcare industry, your main responsibility involves controlling access to sensitive information based on employees' IP addresses. The "allow list" is a roster of IP addresses with granted permission to access specific restricted data, mostly personal patient records. Occasionally, some of these IP addresses must be revoked from accessing such sensitive information. These addresses are stored in a distinct "remove list". Your task involves crafting a Python algorithm to cross-reference the "allow list" with the "remove list", eliminating any shared IP addresses from the "allow list". This protocol ensures that only approved personnel can access the sensitive data.

Procedure as follows.

1. Opening the Allow List File:
Python's built-in open() function is used to open a file. The function requires the filename and mode ('r' for read) as arguments. This operation is typically performed using a 'with' statement, which automatically takes care of closing the file after use.

Example:
```
# Specify filename
import_file = "allow_list.txt"

# Open the file
with open(import_file, 'r') as file:
    allow_list = file.read().split("\n")
```

2. Reading File Contents:
In Python, the read() method reads the content of the file as a single string. The split() function is then used to break up the file content string into a list, with each line as an item in the list.

Example:
```
# Open and read file contents
with open(import_file, 'r') as file:
    allow_list = file.read().split("\n")
```

3. Iterating through the Remove List:
To iterate through the remove list, a for loop in Python is used.

Example:

```
remove_list = ['192.168.1.1', '192.168.1.2', '192.168.1.3']

for ip in remove_list:
    if ip in allow_list:
        allow_list.remove(ip)  # Removes IP from allow_list if found
```

4. Writing Updated IPs to the File:

The updated list of IP addresses (allow_list) needs to be written back to the file. The join() method is used to convert the list back to a string, and the write() method writes the updated list to the file in write mode ('w').

Example:

```
updated_allow_list = '\n'.join(allow_list)

# Open file in write mode and write updated list
with open(import_file, 'w') as file:
    file.write(updated_allow_list)
```

Summary:

This document illustrates a Python algorithm designed to manage access to confidential information in a healthcare environment. This system involves the use of an "allow list" of authorized IP addresses and a separate "remove list" for addresses that need to be revoked. The algorithm involves opening and reading the "allow list" file, iterating through the "remove list", and removing any common IP addresses from the "allow list". The updated "allow list" is then written back to the original file.