Algorithm for file updates in Python

Project description

In this algorithm, we need to update a file containing a list of allowed IP addresses into a network by removing addresses present on a list of addresses marked for removal. In the algorithm below, we first read the allowed IP addresses from the file. After, we remove those that are present on the removal list. Finally, we write the updated list back to the file and read it once again to ensure the file was correctly updated.

Open the file that contains the allow list

```
1   with open('allow_list.txt', 'r') as file:
```

By using 'with', we are able to open the 'allow_list.txt' file and read its contents. 'with' ensures that the file is closed after a successful reading.

Read the file contents

```
1   with open('allow_list.txt', 'r') as file:
2       ip_addresses = file.read()
3   print(ip_addresses)
```

After opening the file, we are able to read and store the contents as a string in the 'ip_addresses' variable. The .read() method allows us to read the contents of the file and convert them into a string.

Convert the string into a list

```
1   with open('allow_list.txt', 'r') as file:
2       ip_addresses = file.read()
3   ip_addresses = ip_addresses.split()
4   print(ip_addresses)
```

After converting the contents into a string and storing them in the 'ip_addresses' variable, we convert the string into a list using the .split() method. This method allows us to split the string into a list based on a provided argument. Since we simply want to split the string by

whitespace, we do not provide an argument. In this case, we have converted the string into a list with 'ip_addresses' now containing the allowed ip addresses in a list.

Iterate through the remove list

```
1  remove_list = ['196.134.34.54', '233.42.32.1']
2  with open('allow_list.txt', 'r') as file:
3      ip_addresses = file.read()
4  ip_addresses = ip_addresses.split()
5  for i in remove_list:
6      print(i)
7  print(ip_addresses)
```

We have now added some ip addresses marked for removal into a 'remove_list' variable present at the top. As shown above, we were able to iterate and print each ip address present in the 'remove_list' variable using a for loop.

Remove IP addresses that are on the remove list

```
1  remove_list = ['196.134.34.54', '233.42.32.1']
2  with open('allow_list.txt', 'r') as file:
3      ip_addresses = file.read()
4  ip_addresses = ip_addresses.split()
5  for i in ip_addresses:
6      if i in remove_list:
7          ip_addresses.remove(i)
8  print(ip_addresses)
```

Next, we used a for loop to iterate through the list of allowed ip addresses and remove them if they are present in the list of addresses marked for removal. The .remove() method allows us to remove the first occurrence of a value that is present in a list.

Update the file with the revised list of IP addresses

```
remove_list = ['196.134.34.54', '233.42.32.1']
with open('allow_list.txt', 'r') as file:
    ip_addresses = file.read()
ip_addresses = ip_addresses.split()
for i in ip_addresses:
    if i in remove_list:
        ip_addresses.remove(i)
ip_addresses = " ".join(ip_addresses)
with open('allow_list.txt', 'w') as file:
    file.write(ip_addresses)
with open('allow_list.txt', 'r') as file:
    ip_addresses = file.read()
print(ip_addresses)
```

Finally, after removing the ip addresses marked for removal from the 'ip_addresses' variable, we need to store this new list back to the 'allow_list.txt' file. We do so by first converting the list of ip addresses back into a string using the .join() method. This allows us to convert a list into a string and join the contents on a specific string value. In our case, we want to again separate the ip addresses by spaces, so we provide a value of " ". Next, we use a with statement again to reopen the 'allow_list.txt' file. However, instead of using 'r' to read the file, we use 'w' to overwrite its contents. We then use the .write() method to write the string stored in the ip_addresses variable to the file. Finally, we use a with statement to open the file once again and read its contents. This allows us to ensure that the file was updated properly.

Summary

We were successfully able to update the contents of 'allow_list.txt' with the correct ip addresses by removing those that were marked for removal. Using a combination of file, string, and list methods such as .read(), .split(), and .remove(), we were able to easily update the file and check if the contents were updated successfully with only a few lines of code.