Algorithm for File Updates in Python

Project Description

In this portfolio activity, I developed an algorithm in Python to update a file that contains an allow list of IP addresses for accessing restricted content in a healthcare company. The algorithm checks if any IP addresses from a remove list are present in the allow list and removes them if necessary. This algorithm ensures that only authorized employees can access sensitive patient records based on their IP addresses.

Open the File that Contains the Allow List

To open the file, I used the with statement and the open() function in Python. The with statement ensures that the file is properly closed after it has been used. I assigned the file name **"allow_list.txt"** to the import_file variable.

Read the File Contents

To read the contents of the file, I used the **.read()** method on the file object. This method reads the entire contents of the file and converts them into a string. I stored this string in a variable called ip_addresses.

Convert the String into a List

To work with individual IP addresses, I used the **.split()** method on the ip_addresses string. This method splits the string into separate IP addresses based on a delimiter (default is a space). I used the default delimiter to split the string into a list of IP addresses.

Iterate through the Remove List

I created a second list called remove_list that contains the IP addresses to be removed from the allow list. To iterate through this list, I used a for loop. The loop variable element represents each IP address in the remove_list.

Remove IP Addresses that are on the Remove List

Within the for loop, I used a conditional statement (if) to check if the current IP address (element) exists in the ip_addresses list. If it does, I used the **.remove()** method on the ip_addresses list to remove that IP address. This ensures that any IP addresses present in the remove list are removed from the allow list.

Update the File with the Revised List of IP Addresses

After removing the necessary IP addresses, I converted the updated ip_addresses list back into a string using the **.join()** method. This method joins the elements of the list into a single string, with each element separated by a specified delimiter. In this case, I used the newline character **"\n"** as the delimiter to separate the IP addresses on separate lines.

Finally, I used another with statement and the **.write()** method on the file object to write the revised allow list back to the file specified by the import_file variable. This ensures that the file is updated with the new list of IP addresses.

Summary

In summary, the algorithm I developed allows for the efficient management of the allow list file in a healthcare company. It checks for IP addresses present in the remove list and removes them from the allow list, ensuring that only authorized employees can access restricted content based on their IP addresses. The algorithm utilizes Python's file handling techniques (with statement, open(), .read(), .write()) as well as list manipulation methods **(.split(), .join())** and control flow structures (for loop, if statement) to accomplish the desired file updates.