

# Algorithm for File Updates in Python

## Project description

Develop an algorithm that parses a file containing IP addresses that are allowed to access restricted content and removes addresses that no longer have access.

## Open the file that contains the allow list

The goal is to develop an algorithm that parses a series of IP addresses that can access restricted information and removes the addresses that are no longer allowed. Python can automate this process.

Given: a text file called "allow\_list.txt" that contains a series of IP addresses that are allowed to access restricted information.

There are IP addresses that should no longer have access to this information, and their IP addresses need to be removed from the text file. You're given a variable named `remove_list` that contains the list of IP addresses to be removed.

```
In [ ]: import_file = "allow_list.txt"
remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]
print(import_file)
print(remove_list)
```

The first line of the output shows the name of the text file. The second line of the output shows the list of IP addresses from the `remove_list`.

## Read the file contents

start by opening the text file using the `import_file` variable, the `with` keyword, and the `open()` function with the "r" parameter.

write the first line of the `with` statement. Running this code will produce an error because it will only contain the first line of the `with` statement;

```
In [ ]: import_file = "allow_list.txt"
remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]
with open(import_file, "r") as file:
    ip_addresses = file.read()
ip_addresses = ip_addresses.split()
print(ip_addresses)
```

## Iterate through the remove list

The code that removes the elements of `remove_list` from the `ip_addresses` list. This will require both an iterative statement and a conditional statement.

First, build the iterative statement. Name the loop variable `element`, loop through `ip_addresses`, and display each element.

```
In [ ]: import_file = "allow_list.txt"
remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]
with open(import_file, "r") as file:
    ip_addresses = file.read()
ip_addresses = ip_addresses.split()
for element in ip_addresses:
    print(element)
```

## Remove IP addresses that are on the remove list

build a conditional statement to remove the elements of `remove_list` from the `ip_addresses` list.

The conditional statement should be placed inside the iterative statement that loops through `ip_addresses`. In every iteration, if the current element in the `ip_addresses` list is in the `remove_list`, the `remove()` method should be used to remove that element.

Afterwards, display the updated `ip_addresses` list to verify that the elements of `remove_list` are no longer in the `ip_addresses`.

```
In [ ]: import_file = "allow_list.txt"
remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]
with open(import_file, "r") as file:
    ip_addresses = file.read()
ip_addresses = ip_addresses.split()
for element in ip_addresses:
    if element in remove_list:
        ip_addresses.remove(element)
print(ip_addresses)
```

## Update the file with the revised list of IP addresses

The next step is to update the original file that was used to create the `ip_addresses` list. A line of code containing the `.join()` method has been added to the code so that the file can be updated. This is necessary because `ip_addresses` must be in string format when used inside the `with` statement to rewrite the file.

The `.join()` method takes in an iterable (such as a list) and concatenates every element of it into a string. The `.join()` method is applied to a string consisting of the character that will be used to separate every element in the iterable once its converted into a string. In

the code below, the method is applied to the string " ", which contains just a space character. The argument of the `.join()` method is the iterable you want to convert, and in this case, that's `ip_addresses`. As a result, it converts `ip_addresses` from a list back into a string with a space between each element and the next.

```
In [ ]: import_file = "allow_list.txt"
remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]
with open(import_file, "r") as file:
    ip_addresses = file.read()
ip_addresses = ip_addresses.split()
for element in ip_addresses:
    if element in remove_list:
        ip_addresses.remove(element)
ip_addresses = " ".join(ip_addresses)
with open(import_file, "w") as file:
    file.write(ip_addresses)
```

Write another with statement, this time to read in the updated file. Start by opening the file. Then read the file and store its contents in the text variable. Afterwards, display the text variable to examine the result.

```
In [ ]: import_file = "allow_list.txt"
remove_list = ["192.168.97.225", "192.168.158.170", "192.168.201.40", "192.168.58.57"]
with open(import_file, "r") as file:
    ip_addresses = file.read()
ip_addresses = ip_addresses.split()
for element in ip_addresses:
    if element in remove_list:
        ip_addresses.remove(element)
ip_addresses = " ".join(ip_addresses)
with open(import_file, "w") as file:
    file.write(ip_addresses)
with open(import_file, "r") as file:
    text = file.read()
print(text)
```

Define a function named `update_file()` that takes in two parameters. The first parameter is the name of the text file that contains IP addresses (call this parameter `import_file`). The second parameter is a list that contains IP addresses to be removed (call this parameter `remove_list`).

```
In [ ]: def update_file(import_file, remove_list):
    with open(import_file, "r") as file:
        ip_addresses = file.read()
    ip_addresses = ip_addresses.split()
    for element in ip_addresses:
        if element in remove_list:
            ip_addresses.remove(element)
    ip_addresses = " ".join(ip_addresses)
    with open(import_file, "w") as file:
        file.write(ip_addresses)
```

Finally, call the `update_file()` that you defined. Apply the function to "allow\_list.txt" and pass in a list of IP addresses as the second argument.

Use the following list of IP addresses as the second argument:

```
["192.168.25.60", "192.168.140.81", "192.168.203.198"]
```

After the function call, use a with statement to read the contents of the allow list. Then display the contents of the allow list. Run it to verify that the file has been updated by the function.

```
In [ ]: def update_file(import_file, remove_list):
        with open(import_file, "r") as file:
            ip_addresses = file.read()
            ip_addresses = ip_addresses.split()
            for element in ip_addresses:
                if element in remove_list:
                    ip_addresses.remove(element)
            ip_addresses = " ".join(ip_addresses)
        with open(import_file, "w") as file:
            file.write(ip_addresses)
update_file("allow_list.txt", ["192.168.25.60", "192.168.140.81", "192.168.203.198"])
with open("allow_list.txt", "r") as file:
    text = file.read()
print(text)
```

## Summary

- Python has functions and syntax that help you import and parse text files.
  - The with statement allows you to efficiently handle files.
  - The open() function allows you to import or open a file. It takes in the name of the file as the first parameter and a string that indicates the purpose of opening the file as the second parameter.
    - Specify "r" as the second parameter if you're opening the file for reading purposes.
    - Specify "w" as the second parameter if you're opening the file for writing purposes.
  - The .read() method allows you to read in a file.
  - The .write() method allows you to append or write to a file.
- You can use a for loop to iterate over a list.
- You can use an if statement to check if a given value is in a list and execute a specific action if so.
- You can use the .split() method to convert a string to a list.
- You can use Python to compare contents of a text file against elements of a list.