Algorithm for file updates in Python

Project description

At my organization, access to restricted content is controlled with an allowed list of IP addresses. The `"allow_list.txt"` file identifies these IP addresses. A separate remove list identifies IP addresses that should no longer have access to this content. I created an algorithm to automate updating the `"allow_list.txt"` file and remove these IP addresses that should no longer have access.

```python
import_file = "allow_list.txt"

def update_file(import_file, remove_list):

  with open(import_file, "r") as file:

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

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

  ip_addresses = ip_addresses.split()

  for element in ip_addresses:

    if element in remove_list:

      # then current element should be removed from `ip_addresses`
      ip_addresses.remove(element)

  # Convert `ip_addresses` back to a string so that it can be written into the text file

  ip_addresses = " ".join(ip_addresses)

  # Build `with` statement to rewrite the original file

  with open(import_file, "w") as file:

    # Rewrite the file, replacing its contents with `ip_addresses`

    file.write(ip_addresses)

remove_list = ["192.168.25.60", "192.168.140.81", "192.168.203.198"]

update_file(import_file, remove_list)

# Build `with` statement to read in the updated file

with open("allow_list.txt", "r") as file:

  # Read in the updated file and store the contents in `text`
  text = file.read()

print(text)
```

Function: `update_file(import_file, remove_list)`
Purpose: This function removes specified IP addresses from a text file named `"allow_list.txt"`.

Parameters:

`import_file`: (str) The path to the file to be updated. Defaults to `"allow_list.txt"`.
`remove_list`: (list) A list of IP addresses to be removed from the file.

Steps:

1. **Open the file:** The function starts by opening the `import_file` in read mode using a `with` statement.
2. **Read the file contents:** The entire file content is read into a variable named `ip_addresses` using the `read()` method.
3. **Convert to a list:** The `ip_addresses` string is split into a list of individual IP addresses using the `split()` method.
4. **Iterate and remove unwanted IPs:** A loop iterates through each element in the `ip_addresses` list. If the element (IP address) is present in the `remove_list`, it is removed from the `ip_addresses` list using the `remove()` method.
5. **Convert back to string:** The updated `ip_addresses` list is converted back into a string with each element separated by a space using the `join()` method.
6. **Rewrite the file:** The `import_file` is opened again in write mode using another with statement. The updated `ip_addresses` string is written back to the file using the `write()` method, effectively replacing the original content.
7. **(Optional) Read and display updated file:** The script demonstrates reading the updated file content and displaying it using another `with` statement and `read()/print()` methods.

Summary:

The `update_file` function provides a convenient way to remove unwanted IP addresses from a text file. It iterates through the file content, removes specified entries, and rewrites the entire file with the updated list. This function can be used to maintain a whitelist or blacklist of IP addresses for various security or access control purposes.

Additional notes:
- The function assumes the provided remove_list contains valid IP addresses.
- Error handling could be added to address potential issues like invalid file paths or invalid IP addresses in the list.