

PE03: Programming Exercise

Instructions

[shopping_list.py](#)

Description

This assignment is to gain knowledge on Stack and Recursive functions.

A shopping list using Stack will be implemented for this exercise.

The program stores shopping items in both of simple array or linked list based Stack.

Each storing mechanism is separated into different files, such as

[simple_array_shopping_list_manager.py](#) or [linked_list_shopping_list_manager.py](#), where each file contains `FileNameClass` classes with essential methods for data manipulation.

Note that [shopping_list.py](#) with the `main` method has already been provided (download attachment).

As part of the assignment, compare the actual runtime of insert operation between two lists, and justify in a short paragraph on how the algorithms perform.

Keep in mind to always comment and document your class and methods.

Documentation Reference

Mertz, J. (2018). Documenting python code: A complete guide. [realpython.com](#).

<https://realpython.com/documenting-python-code/>

Expected Result

[shopping_list.py](#)

This is the Main python file which is already provided and contains main and test procedure which calls methods implemented on

"[simple_array_shopping_list_manager.py](#)" or "[linked_list_stack_shopping_list_manager.py](#)" file to manage shopping items.

(This is already provided, but please include this file on your submission).

[simple_array_shopping_list_manager.py](#)

This is a file that includes the class of simple array-based shopping list manager.

This class contains such methods as `init`, `insert_item`, `print_items`.

Please keep in mind the following notes for each method during implementation:

- `Init()`: initializes simple array to be used throughout object life.
- `insert_item(item)`: inserts item at the front of the array.
 - **Parameters:** item name.
 - **Note** that you are allowed to use the `insert()` method from Python Array Module.

- `print_items()`: simply prints item.

`linked_list_stack_shopping_list_manager.py`

This is a file that includes the class of linked list based shopping list manager.

This class contains such methods as:

1. `init`
2. `insert_item`
3. `is_list_empty`
4. `print_item_recursive_from_top`
5. `print_items_from_top`
6. `print_item_recursive_from_bottom`
7. `print_items_from_bottom`
8. `getLastItem`
9. `removeLastItem`.

In addition, this class requires the inner class to hold onto data as a linked list based on Stack.

DO NOT use standard python library, such as `deque` from the collections module.

Please keep in mind the following notes for each method during implementation:

- `Init()`: initializes linked list based on Stack object to be used throughout object life.
- `insert_item(item)`: inserts item at the front of the linked list based on Stack.
 - **Parameters:** item name.
- `is_list_empty()`: checks if the current Stack object is empty (ex. check if head is None).
- `print_item_recursive_from_top(currentNode)`: a helper method to print linked list based on Stack item recursively.
 - **Note:** try to print as [`itemTop itemTop-1 item...`] from the top of the Stack by using some combinations of "`print(item, end = " ")`".
 - **Parameters:** current visiting node.
- `print_items_from_top()`: calls `print_item_recursive_from_top` method with current Stack object.
- `print_item_recursive_from_bottom(currentNode)`: a helper method to print linked list based Stack item recursively.
 - **Note:** try to print as [`itemBottom itemBottom+1 item...`] from the bottom of the Stack by using some combinations of "`print(item, end = " ")`".
 - **Parameters:** current visiting node.
 - **Hint:** try to rearrange when you use the print method.
- `print_items_from_bottom()`: calls `print_item_recursive_from_bottom` method with current Stack object.
- `getLastItem()`: returns last inserted item data. This method operates similar as Stack Peek operation.
- `removeLastItem()`: returns the last inserted item while removing it. This method operates similar as Stack Pop operation.

Comparison

As part of the assignment, compare the actual runtime of insert operation between two lists and justify in a short paragraph on how they perform.

Comparison WriteUp

Comparison of the actual runtime of the insert operation between the two lists.

1. The `simple_array_shopping_list_manager.py` implementation will have a longer runtime than the `linked_list_stack_shopping_list_manager.py` implementation.
2. This is because the insert operation for the simple array implementation has a time complexity of $O(n)$ in the worst case
 1. it may need to shift all the elements when inserting a new item at the beginning of the list
3. The insert operation for the linked list implementation has a time complexity of $O(1)$
 1. It only needs to update the head pointer and the next pointer of the new node.
4. In general, linked lists perform better than simple arrays for insert operations at the beginning of the list, as they do not require shifting any elements

Example Output

```
-insert: 1.0199990356341004e-05
-lookup: 4.00003045797348e-07
-insert: 1.0199990356341004e-05

----- Linked list -----
current list:  fish mushroom beef pork carrot cheese butter bread milk banana apple
last element:  fish
-insert: 1.2499978765845299e-05
-lookup: 4.699977580457926e-06
-insert: 1.2499978765845299e-05

VL> & "C:/Users/Thaddeus Maximus/AppData/Local/Programs/Python/Python311/python.exe" d:/Repositories/PEs/Algorithms/VL/modules/PE03/shopping_lst.py
----- Simple array -----
current list:  fish mushroom beef pork carrot cheese butter bread milk banana apple
Sorted(a to z): fish mushroom beef pork carrot cheese butter bread milk banana apple
-insert: 1.090002479031682e-05
-lookup: 4.00003045797348e-07
-insert: 1.090002479031682e-05

----- Linked list -----
current list:  fish mushroom beef pork carrot cheese butter bread milk banana apple
last element:  fish
-insert: 1.1099997209385037e-05
-lookup: 4.4000043999403715e-06
-insert: 1.1099997209385037e-05

VL> & "C:/Users/Thaddeus Maximus/AppData/Local/Programs/Python/Python311/python.exe" d:/Repositories/PEs/Algorithms/VL/modules/PE03/shopping_lst.py
----- Simple array -----
current list:  fish mushroom beef pork carrot cheese butter bread milk banana apple
Sorted(a to z): fish mushroom beef pork carrot cheese butter bread milk banana apple
-insert: 1.0199990356341004e-05
-lookup: 2.999731805175543e-07
-insert: 1.0199990356341004e-05

----- Linked list -----
```