PE04: Programming Exercise

Instructions

shopping_list.py

Description

This assignment is to gain knowledge on quicksort using a linked list. A shopping list will be reimplemented for this exercise. The program stores shopping items in both simple array or linked list. 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 quicksort 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.

Reference

Mertz, J. (2022). Documenting Python Code: A Complete Guide.

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_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`, `delete_item`, `get_last_item`, `selection_sort`. 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.
- `quick_sort_helper(array)`: since the item list is within the class, this helper method takes in the array and runs quick sort recursively.
- **Parameters**: array item list to be sorted.
- `quick_sort()`: sorts items on the array from the current object and replaces it with the newly sorted result.

linked_list_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 `init`, `insert_item`, `print_items`, `delete_item`, `get_last_item`, `selection_sort`. In addition, this class requires the inner class to hold onto data as a linked list. Please keep in mind the following notes for each method during implementation:

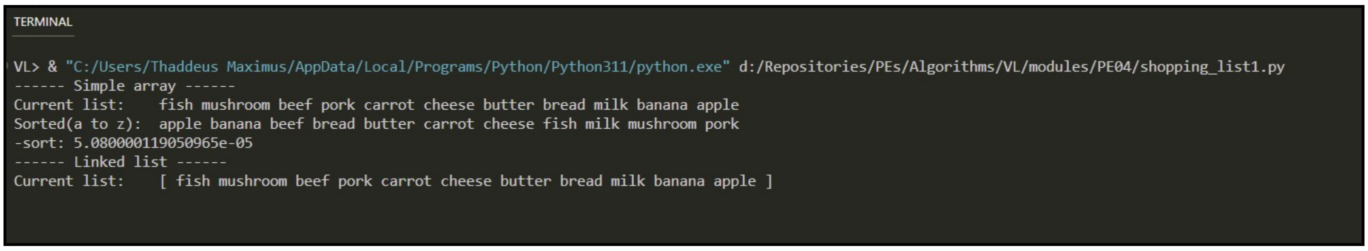
- `Init()`: initializes linked list object to be used throughout object life.
  `insert_item(item)`: inserts item at the front of the linked list.
- **Parameters**: item name.
- `print_items()`: simply prints items throughout the linked list.
- ***Note***: try to print as `[ item1 item2 ]` by using some combinations of "`print(item, end = " ")`".
- `quick_sort_helper(linked_list)`: since the item list is within the class, this helper method takes in the linked list and runs quick sort recursively.
- **Parameters**: linked list item list to be sorted.
- `quick_sort()`: sorts items on the linked list from the current object and replaces it with the newly sorted result.

Comparison

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

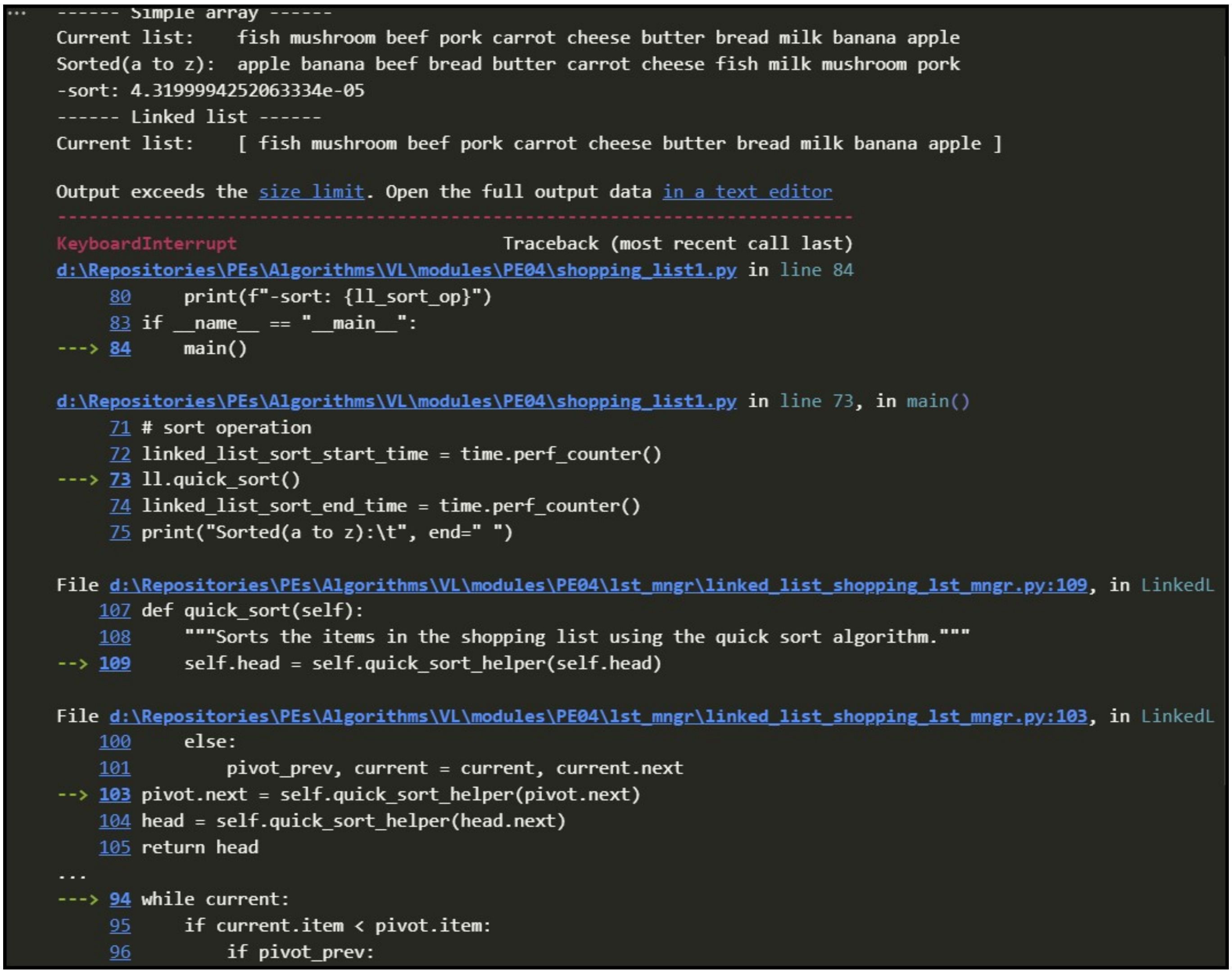
As you can see in the example below the comparison doesnt work as expected because of a stack overflow. I ran out of time trying to fix it and never got my test modules to work either.

Screenshots



```
TERMINAL

VL> & "C:/Users/Thaddeus Maximus/AppData/Local/Programs/Python/Python311/python.exe" d:/Repositories/PEs/Algorithms/VL/modules/PE04/shopping_list1.py
------ Simple array ------
Current list:    fish mushroom beef pork carrot cheese butter bread milk banana apple
Sorted(a to z):  apple banana beef bread butter carrot cheese fish milk mushroom pork
-sort: 5.080000119050965e-05
------ Linked list ------
Current list:    [ fish mushroom beef pork carrot cheese butter bread milk banana apple ]
```



```
...  ------ Simple array ------
     Current list:    fish mushroom beef pork carrot cheese butter bread milk banana apple
     Sorted(a to z):  apple banana beef bread butter carrot cheese fish milk mushroom pork
     -sort: 4.3199994252063334e-05
     ------ Linked list ------
     Current list:    [ fish mushroom beef pork carrot cheese butter bread milk banana apple ]

     Output exceeds the size limit. Open the full output data in a text editor
     ---------------------------------------------------------------------------
     KeyboardInterrupt                         Traceback (most recent call last)
     d:\Repositories\PEs\Algorithms\VL\modules\PE04\shopping_list1.py in line 84
          80     print(f"-sort: {ll_sort_op}")
          83 if __name__ == "__main__":
     ---> 84     main()

     d:\Repositories\PEs\Algorithms\VL\modules\PE04\shopping_list1.py in line 73, in main()
          71 # sort operation
          72 linked_list_sort_start_time = time.perf_counter()
     ---> 73 ll.quick_sort()
          74 linked_list_sort_end_time = time.perf_counter()
          75 print("Sorted(a to z):\t", end=" ")

     File d:\Repositories\PEs\Algorithms\VL\modules\PE04\lst_mngr\linked_list_shopping_lst_mngr.py:109, in LinkedL
         107 def quick_sort(self):
         108     """Sorts the items in the shopping list using the quick sort algorithm."""
     --> 109     self.head = self.quick_sort_helper(self.head)

     File d:\Repositories\PEs\Algorithms\VL\modules\PE04\lst_mngr\linked_list_shopping_lst_mngr.py:103, in LinkedL
         100     else:
         101         pivot_prev, current = current, current.next
     --> 103 pivot.next = self.quick_sort_helper(pivot.next)
         104 head = self.quick_sort_helper(head.next)
         105 return head
     ...
     ---> 94 while current:
          95     if current.item < pivot.item:
          96         if pivot_prev:
```