Results

Return

This is the return from running shopping_list.py

```
------ Simple array ------
current list:    ['fish', 'mushroom', 'beef', 'pork', 'carrot', 'cheese',
'butter', 'bread', 'milk', 'banana', 'apple']
delete milk:     ['fish', 'mushroom', 'beef', 'pork', 'carrot', 'cheese',
'butter', 'bread', 'banana', 'apple']
last element:   apple

Sorted(a to z):  ['apple', 'banana', 'beef', 'bread', 'butter', 'carrot',
'cheese', 'fish', 'mushroom', 'pork']
-insert: 2.750000567175448e-05
-delete: 4.4000043999403715e-06
-lookup: 2.300017513334751e-06

------ Linked list ------
current list:    [ fish mushroom beef pork carrot cheese butter bread milk banana
apple ]
delete milk:     [ fish mushroom beef pork carrot cheese butter bread banana apple
]
last element:   apple

smallest element:       apple

Sorted(a to z):  [ apple banana beef bread butter carrot cheese fish mushroom pork
]
smallest element:       apple

-insert: 1.4599994756281376e-05
-delete: 6.89999433234334e-06
-lookup: 3.5999983083456755e-06
```

The results of running shopping_list.py show the comparison between the simple array implementation and the linked list implementation of a shopping list manager.

simple array implementation

1. Insert operation took approximately 2.75e-05 seconds.
2. Delete operation took approximately 4.4e-06 seconds.
3. Lookup operation took approximately 2.3e-06 seconds.

linked list implementation

1. Insert operation took approximately 1.46e-05 seconds.

2. Delete operation took approximately 6.9e-06 seconds.
3. Lookup operation took approximately 3.6e-06 seconds.

Observations

- The insert operation is slightly faster in the linked list implementation compared to the simple array implementation.
- The delete operation is faster in the simple array implementation compared to the linked list implementation. This is because the linked list implementation has to traverse the list until it finds the item to delete, whereas the simple array implementation can directly access the item.
- The lookup operation is faster in the simple array implementation compared to the linked list implementation, as the array allows for direct access to the elements, while the linked list needs to traverse the list to find the desired item.

It is important to note that these results are specific to this particular case and may not be generalizable to all use cases.
The performance of each implementation depends on various factors such as the size of the list, the distribution of the data, and the frequency of different operations.

Smallest implementation

The smallest element in the shopping list are determined & sorted alphabetically. In this case, "apple" comes before all the other items in the list alphabetically, so it is considered the smallest element.

Comparing "apple" with "beef" or "pork", 'a' comes before 'b' and 'p', making "apple" the smallest element in the list. This has nothing to do with the length of the words themselves.