

Ismael Villalobos
February 27, 2019
ID:80586103
Professor Fuentes

Lab 2

Introduction: The purpose of this assignment is to implement several algorithms for finding the median of a list of integers, using objects of the List class described in class, and compare their running times (measured as the number of comparisons each algorithm makes) for various list lengths.

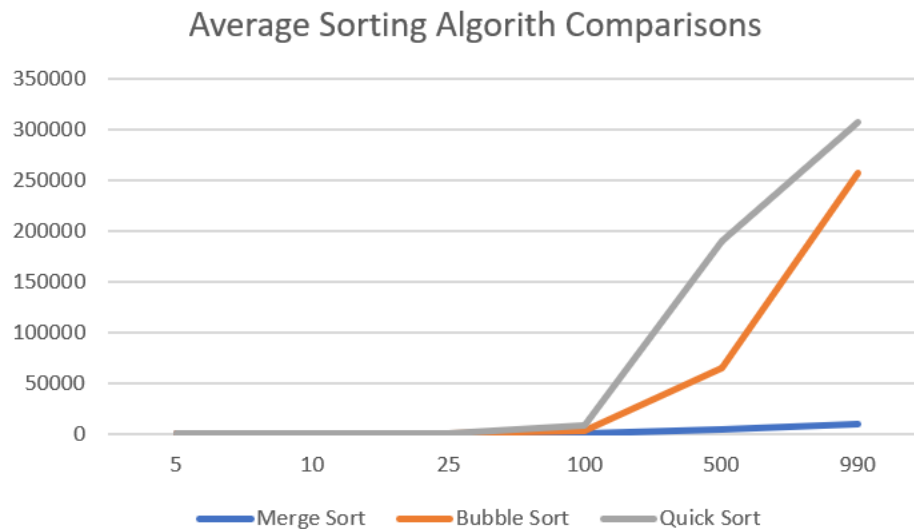
Procedure:

For testing purposes I ran my code 5 times with said n number of values.

Sample Terminal output

```
Unsorted List
967 235 22 869 187 403 780 514 596 350
Sorted using Bubble Sort
22 187 235 350 403 514 596 780 869 967
Number of Comparisons using Bubble Sort 23
Median using Bubble Sort is 514
Sorted using Quick Sort
22 187 235 350 403 514 596 780 869 967
Number of Comparisons using Quick Sort 45
Median using Quick Sort is 514
Sorted using Merge Sort
22 187 235 350 403 514 596 780 869 967
Number of comparisons using Merge sort 34
Median using Merge Sort is 514
Median using Modified Quick Sort is 514
```

Ismael Villalobos
February 27, 2019
ID:80586103
Professor Fuentes
Results



Lists of Size n	Average number of Comparisons		
	Merge Sort	Bubble Sort	Quick Sort
5	12	4.8	10
10	34	24.6	45
25	118	147.6	300
100	672	2306.2	4960
500	4488	60975.6	124750
990	9866	248130.4	49555

Conclusion

In conclusion we were able to create sorting algorithms for linked lists of various sizes. We then took the outputs and created graphs to visualize the speed of sorting algorithms.

Appendix

Source Code provided by Professor Fuentes, other parts were created for lab assignment

```
class Node(object):
    # Constructor
    def __init__(self, item, next=None):
        self.item = item
        self.next = next

def PrintNodes(N):
    if N != None:
        print(N.item, end=' ')
        PrintNodes(N.next)
```

Ismael Villalobos
February 27, 2019
ID:80586103
Professor Fuentes

```
def PrintNodesReverse(N):  
    if N != None:  
        PrintNodesReverse(N.next)  
        print(N.item, end=' ')
```

```
# List Functions  
class List(object):  
    # Constructor  
    def __init__(self):  
        self.head = None  
        self.tail = None  
        self.Len = 0
```

```
def IsEmpty(L):  
    return L.head == None
```

```
def Append(L, x):  
    # Inserts x at end of list L  
    if IsEmpty(L):  
        L.head = Node(x)  
        L.tail = L.head  
        L.Len += 1  
    else:  
        L.tail.next = Node(x)  
        L.tail = L.tail.next  
        L.Len += 1
```

```
def Prepend(L, x):  
    ##inserts x at begingin of list L  
    if IsEmpty(L):  
        L.head = Node(x)  
        L.tail = L.head  
        L.Len += 1  
    else:  
        L.head = Node(x, L.head)  
        L.Len += 1
```

```
def Print(L):  
    # Prints list L's items in order using a loop  
    temp = L.head  
    while temp is not None:  
        print(temp.item, end=' ')  
        temp = temp.next  
    print() # New line
```

Ismael Villalobos

February 27, 2019

ID:80586103

Professor Fuentes

def Remove(L, x):

Removes x from list L

It does nothing if x is not in L

if L.head == None:

return

if L.head.item == x:

if L.head == L.tail: # x is the only element in list

L.head = None

L.tail = None

L.Len -= 1

else:

L.head = L.head.next

L.Len -= 1

else:

Find x

temp = L.head

while temp.next != None and temp.next.item != x:

temp = temp.next

if temp.next != None: # x was found

if temp.next == L.tail: # x is the last node

L.tail = temp

L.tail.next = None

L.Len -= 1

else:

temp.next = temp.next.next

L.Len -= 1

def Copy(L):

copies list L into a new list with the same values

copy = List()

temp = L.head

while temp != None:

Append(copy, temp.item)


t = temp.next

return copy

Ismael Villalobos
February 27, 2019
ID:80586103
Professor Fuentes

Academic Honesty Form for Students

"I understand that to protect the value of the independent work that I do in this course, the work of all students in the course may be compared for evidence of plagiarism to the work of other students, both in this course and in others, and to other sources on the Internet and elsewhere; this may involve the storage of students' work on computer systems outside of the university. I understand that my instructor will be the only person who can see my work so stored. I promise to follow all the university, departmental, and course policies about academic honesty."

Signature:  _____

Date: Feb 27, 2018

Printed name: Ismael Villalobos