Homework #3 Insertion Sort

CSE 1384 – Intermediate Computer Programming

Section - 02

Homework #3 Report

Instructor: Mrs. Jankun-Kelly

RAMESH SHRESTHA

NETID: RS2401

Lab TA: Delma Nieves-Rivera

"On my honor, as a Mississippi State University student, I have neither given nor received unauthorized assistance on this academic work."

Algorithm Design

set n= length of the unsorted list
create an empty sorted list
put the first value of the unsorted list into the sorted list
for each index i in the unsortedlist:
    value= unsortedlist[i]
    for each index j in sortedlist:
        number= sortedlist[j]
        if value is less than number of the sorted list:
            insert value into the sorted list at index j
            break out of the loop
        else if value if greater than the maximum value of the sorted list:
            add the value at the end of the sorted list
            break out of the loop
return the sorted list

Algorithm Analysi

1. What is n?
   n is the length of the list.

2. What is C?
   C is the time to do one comparison with one other element of the list.

3. What is the run time f(n)? How did you derive it?
   We have to do 1+2+……..(n-2)+(n-1) comparisons
   So,
   f(n) =C. $\sum_{i=1}^{n-1} i$
   =C. $\frac{(n-1)\{(n-1)+1\}}{2}$
   =$C.\ \frac{n(n-1)}{2}$
   =$\frac{Cn^2}{2} - \frac{Cn}{2}$

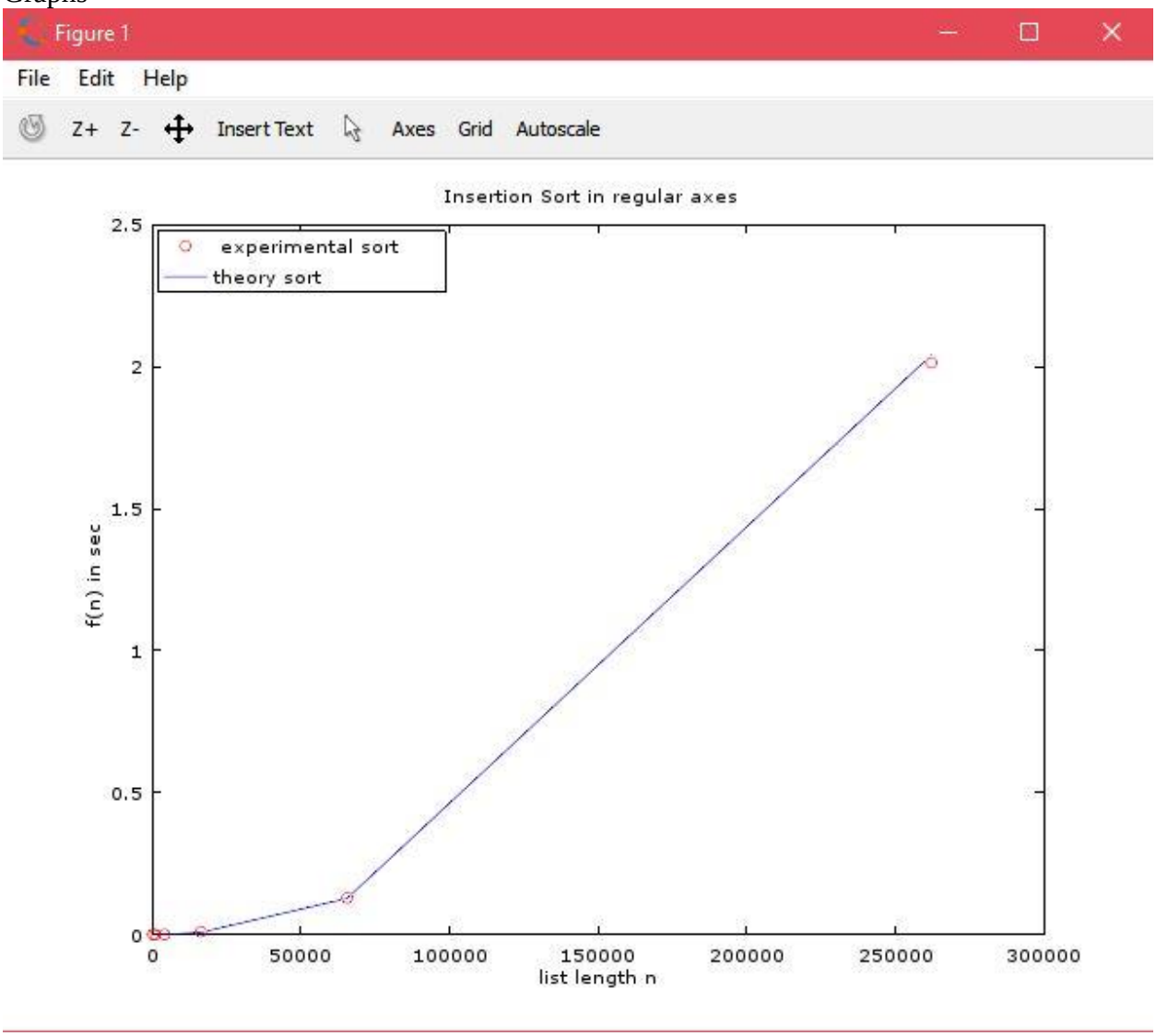
4. What is the Big O upper bound
   $O(n^2)$

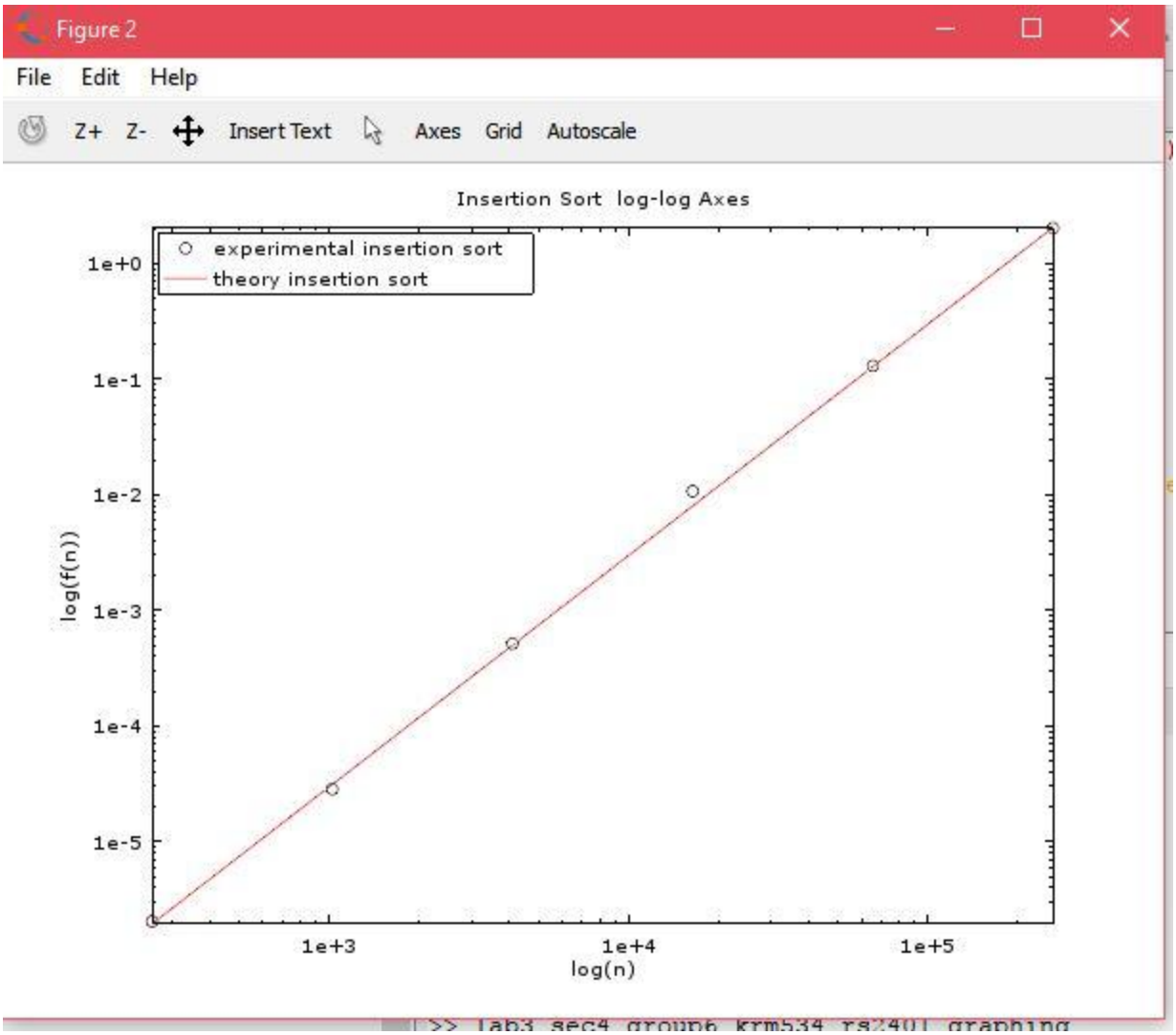
Timing Table

Sample size (n)	Experimental sort time in sec	C values in seconds
256	0.00000204	0.0000000000144
1024	0.00002849	0.0000000000243
4096	0.000515	0.0000000000250
16384	0.0107	0.0000000000383
65536	0.13	0.0000000000336
262144	2.014	0.0000000000263

Estimated C value: 0.0000000000297 sec

Graphs





Code Appendix

Python Code

```
#import time, random, and math modules
from time import time
from random import randint
from math import log2
#define a function insertion sort
def insertion_sort(unsorted_list):
    #create an empty sorted list
```

```
    sorted_list=[]
    #add valueues to the sorted list
    sorted_list.append(unsorted_list[0])

    n=len(unsorted_list)
    #check each values of unsorted list with sorted list and add
the valueues to the sorted list.

    for i in range(1,n):
        value= unsorted_list[i]

        for j in range(len(sorted_list)):
            number=sorted_list[j]
            #check if valueue is less than than the number in
the sorted list and insert the valueue to the list

            if value<number:
                sorted_list.insert(j,value)


                break
            #check if the valueue is greater than the valueues
of the sorted list
            elif value>= sorted_list[len(sorted_list)-1]:

                sorted_list.append(value)

                break
    return sorted_list
#create an empty list
mylist=[]
#create an empty list of list sizes
listSize=[256, 1024, 4096, 16384, 65536,262144]
#create the list of serial already sorted list of integers from
0 to listSize specified
for Size in listSize:
    for w in range(0,2**15):
       mylist.append(w)
    z= len(mylist)
    #start timing
    start_time= time()
    #time for larger repetition and call the insertion sort
function
    for r in range(10000):
        listsort = insertion_sort(mylist)
    #end timing
    end_time = time()
```

```
    time_all_reps = end_time - start_time
    #get the single timing
    timeonce = time_all_reps / 10000
    C = timeonce/ (z*z)
# print the results
    print("list length Size: ",z)
    print("single sorting time: ", format(timeonce, " .3g"))
    print("value of C: ", format(C, ".3g"))
```

Matlab Code

```
%enter list sizes
n = [256, 1024, 4096, 16384, 65536,262144];
%put experimental values for timing
exp_time_insertion_sort = [0.00000204, 0.00002849, 0.000515,
0.0107, 0.13, 2.014];
%enter experimental C value
C_insertion_sort=0.0000000000297;
%calculate theoretical insertion sort values
theory_insertionsort = C_insertion_sort* n.^2.;
figure(1)
clf
%plot the data
plot( n, exp_time_insertion_sort, 'ro' )

hold on

plot( n, theory_insertionsort, 'b-' )

hold off

legend( ' experimental sort  ', 'theory sort ', 'Location',
'northwest')
%set x and y axis labels and title
xlabel( ' list length n ' )
ylabel( ' f(n) in sec' )
title( 'Insertion Sort in regular axes  ' )

set( gcf, 'Color', [ 1 1 1 ] )
%figure 2 for log axes
figure(2)
clf
loglog( n, exp_time_insertion_sort, 'ko' )
```

```
hold on
loglog( n, theory_insertionsort, 'r-' )
hold off
%set axis label
xlabel('log(n)')
ylabel('log(f(n))')
%set axis title
title('Insertion Sort  log-log Axes')
%set axis legend

legend( 'experimental insertion sort','theory insertion
sort','Location', 'northwest')

axis tight
```