Gebze Technical University
Computer Engineering

CSE 222
2017 Spring

HOMEWORK HW02 REPORT

BURAK AKTEN
141044045

Course Assistant: NUR BANU ALBAYRAK

1. System Requirements

No need for this homework.

2. Use Case Diagrams

No need for this homework.

3. Class Diagrams

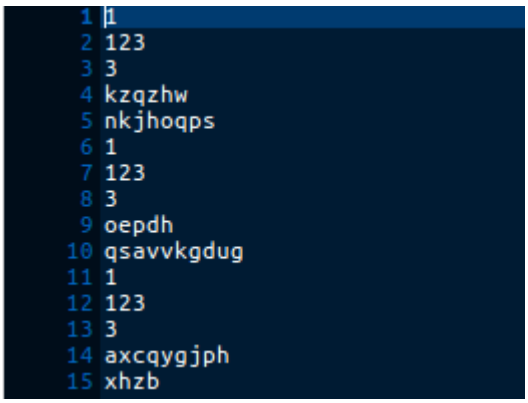
No need for this homework.

4. Other Diagrams

No need for this homework.

5. Problem Solutions Approach

I used input datas in order to compare perfonmance of the codes that is written by using ArrayList and LinkedList. I create five kinds of input files that includes 5000 , 10000 , 20000 , 30000 inputs according to the hierarchy of the code. You can see one example of the files on below . By using this input files I take inputs from file instead of user in loop . Thanks to that, I could compare their running times. To determine the running times of these codes , I used “time” command on Linux terminal.



This is a small part of a input file.
Inputs:

1 for staff loggin.
123 staff password.
3 for registering new user .
First string for book name.
Second string for author name

5000 inputs mean register 1000 user.
10000 inputs mean register 2000 user.
20000 inputs mean register 4000 user.
30000 inputs mean register 6000 user.

6. Test Cases

In my code there are 5 different main metod for system users. They are adding book , removing book and registering new user for staff , borrowing book and returning book for user . To test my code running times with ArrayList and LinkedList I just use registering new user for staff. Before testing , I expected that the code is written by using LinkedList would be less efficent than ArrayList. After testing , I saw that it is like I was thought. I did not test my code with array because I thought that ArrayList was created by using array list. So the efficience would be same like ArrayList.

7. Running and Results

Running Times		Total inputs from file
LinkedList	ArrayList	
2.455 seconds	1.592 seconds	5000
8.108 seconds	3.652 seconds	10000
55.857 seconds	10.275 seconds	20000
2 minute 57,550 seconds	21.768 seconds	30000

As you can see above , when I run the code with small input data , we can not see big differences between two codes(with ArrayList , with LinkedList). But the input data is not small then the perfonmance of the code with ArrayList is greater than other one.

Questions

1.

Q1)

1)
```
for (int i=0; i<n-1; i++){
    for( int j=i+1; j<n ; j++){
        3 simple statements
    }
}
```

when i = 0 (n-1) times
inner loop => j = 1 ——— j = n-1
when i = 1
inner loop => j = 2 (n-2) times ——— j = n-1
⋮
when i = n-2
inner loop => j = n-1 (1) times ——— j = n-1

total
$1+2+\cdots+(n-1)$
$\frac{(n-1).n}{2}$

$T(n) = O\left(\left(3.\frac{n.(n-1)}{2}\right)\right)$

3 for 3 simple statements

$T(n) = O(1.5(n.(n-1)))$ → Drop constant and ignore lower order terms

$T(n) = O(n^2)$

2)
```
public static int length(String str){
    if (str == null || str.equals(""))
        return 0;
    else
        return 1 + length(str.substring(1));
}
```

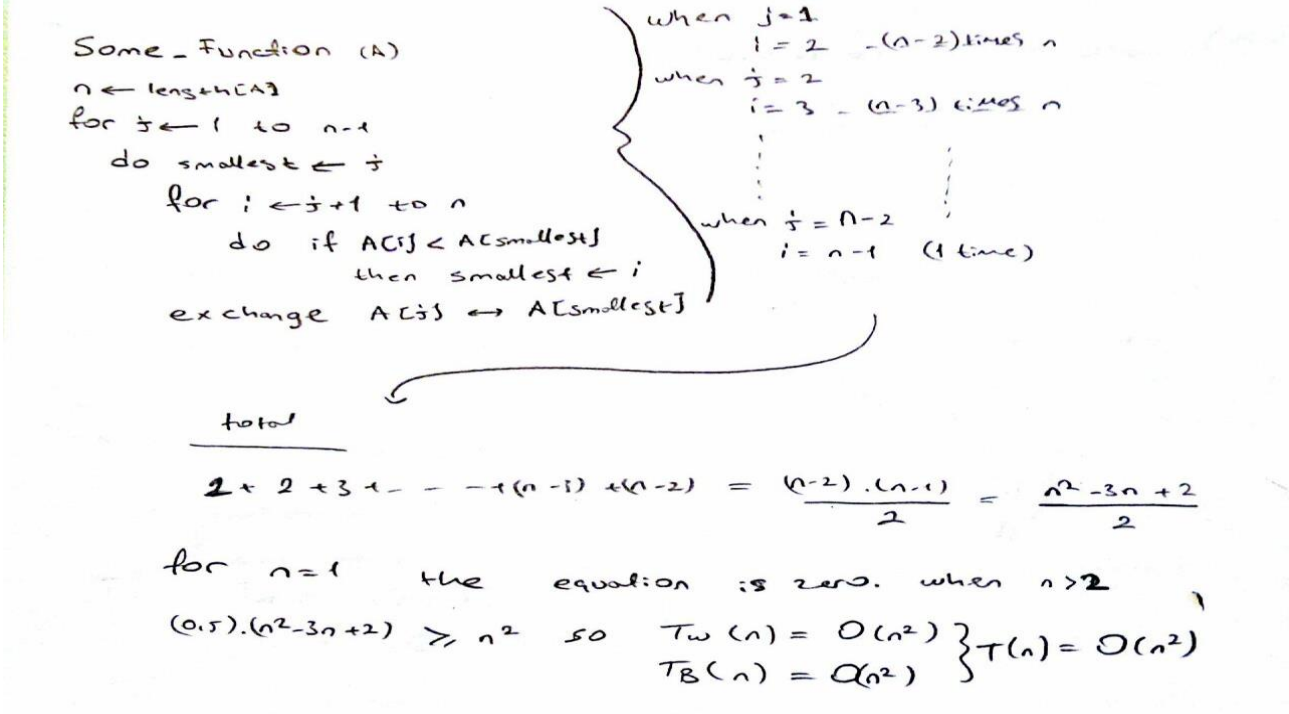
There is two condition so, for best one
"str == null" and "str.equals("")" are constant time
so $T_B(n) = O(1)$
$= \Theta(1)$
$= \Omega(1)$, if first condition true.

Otherwise, lets say the length of str is "n."

length.(str.substring(1)) — (n-1) times
(n-1) times

So for worst one
$T_w(n) = O(2n-2)$
$T_w(n) = O(n)$

2. This function makes sorting from small number to large number.



3.

T(n) = Θ(g(n)) >> This equation means that :

- c' * g(n) >= T(n) >= c'' * f(n) where n >= n0

From this equation : the worst case running time is

- T(n) = O(g(n)) from c' * g(n) >= T(n) , the best case running time is

- T(n) = Ω(g(n)) from T(n) >= c'' * f(n).

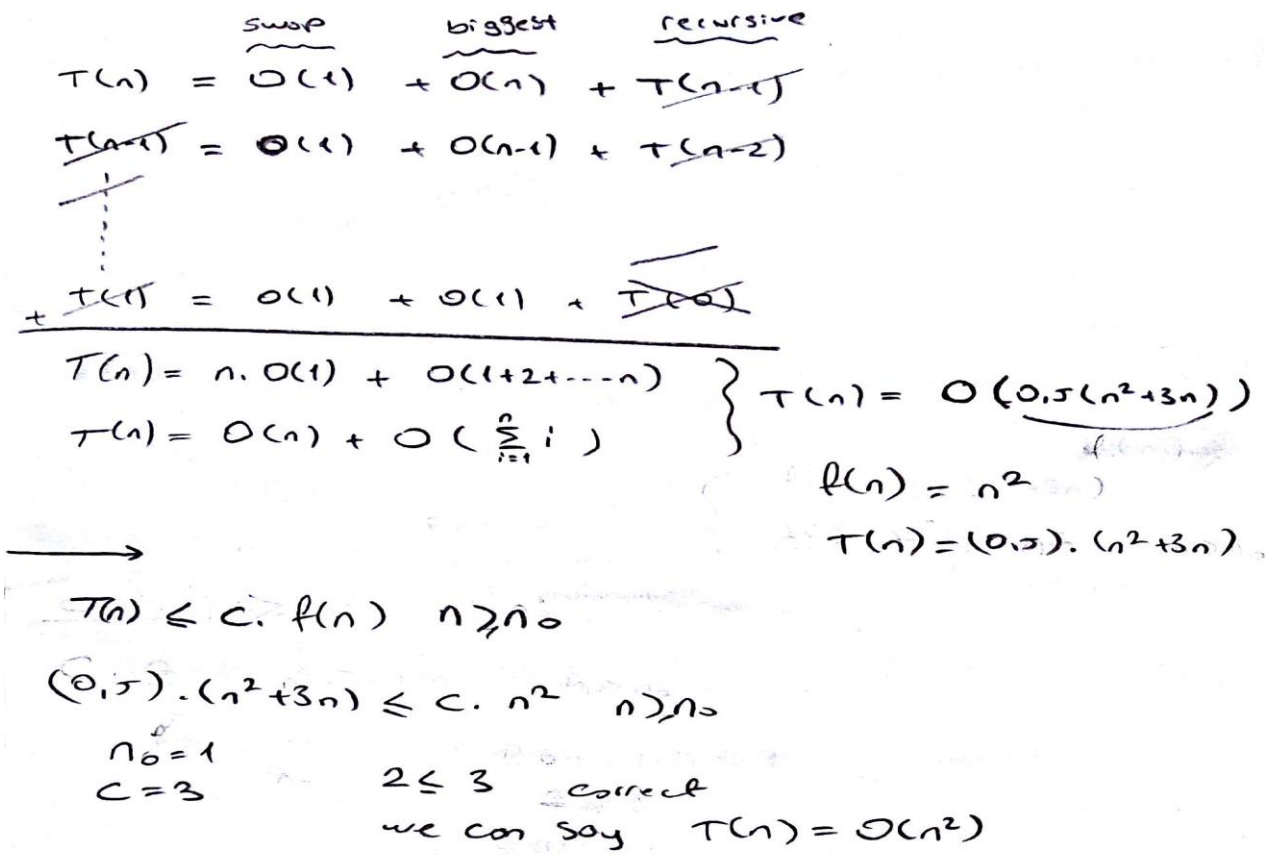
4.

➢

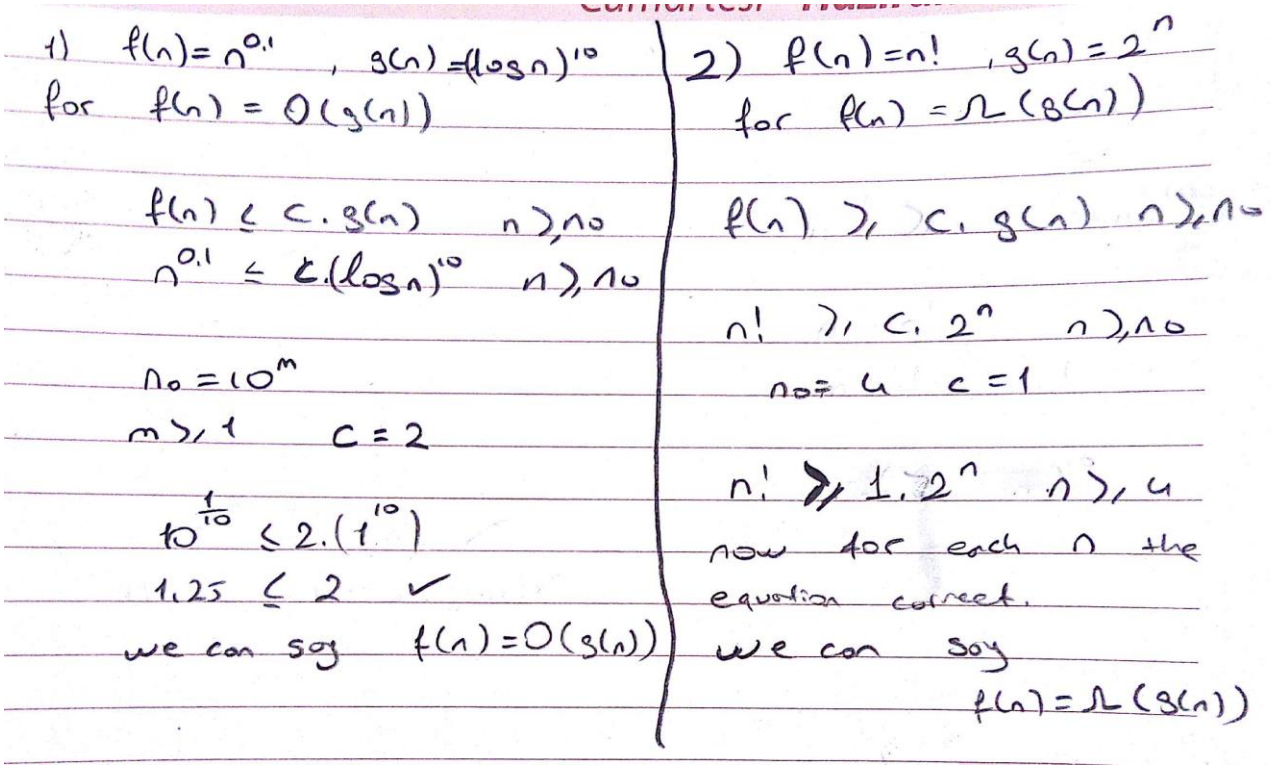
```
public  static void insertion_sort(int arr[] , int size)throws Exception{
        if(size == 0)
                throw new Exception();
        if(size == 1) return;
        else{
                int index = biggest(arr , size);
                swap(arr , size - 1 , index );
                insertion_sort(arr , size  - 1);
        }
}
```

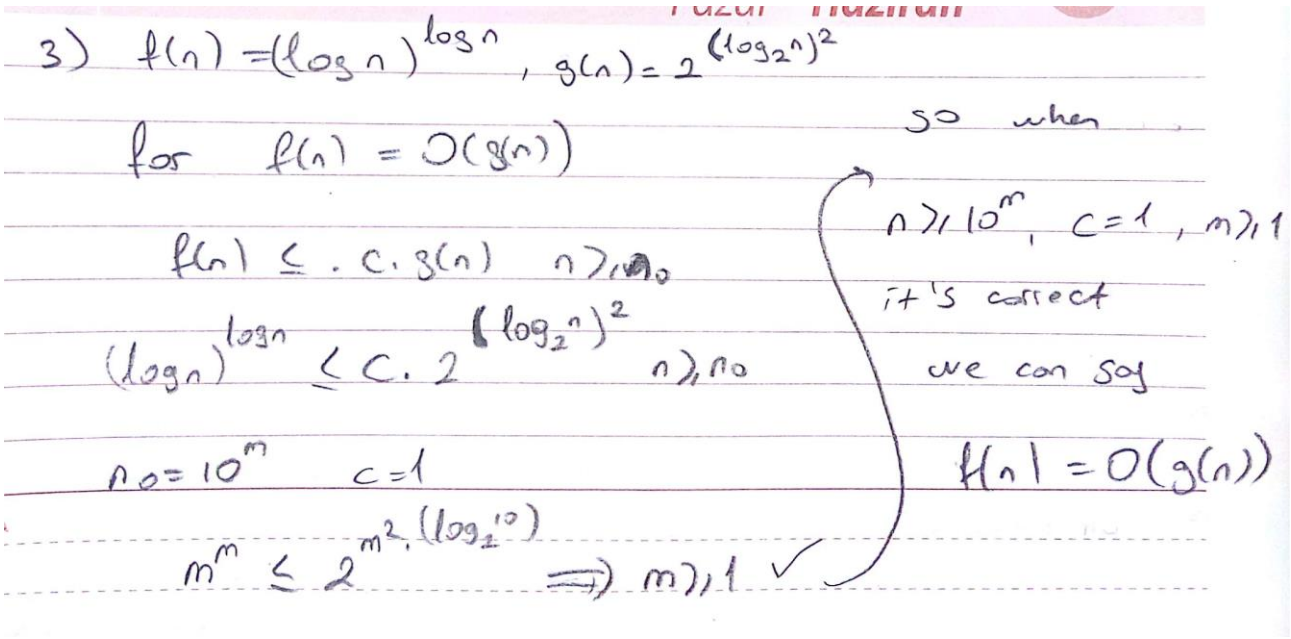
I've write this recursive insertion sort code and I used two helper function in it. First one is swap function that includes 3 simple statements, that is T(n)=O(n)= Θ(n) and other one is biggest that the running time(T(n)) is equal to O(n).

- The recurrence of the worst-case running time and solution.



5.





6.

For this question, I made explanation above in running and results part.