Gebze Technical University
Computer Engineering

CSE 222 - 2018 Spring

HOMEWORK 4 REPORT

Jwan hussein
151044078

Course Assistant:

1.

a)

```
public static Iterator<Integer> longest_sorted_it(LinkedList<Integer> L){
    LinkedList<Integer> s = new LinkedList<Integer>();
    LinkedList<Integer> t = new LinkedList<Integer>();
    Iterator<Integer> it = L.iterator();
    int i;
    boolean is_sorted = true ;
    while (it.hasNext()){
        i = it.next();
        if ( !s.isEmpty() && i < s.getLast()){
            if(s.size() > t.size())
                t = new LinkedList<Integer>(s);
            s.clear();
            is_sorted = false ;
        }
        s.add(i);
    }
    if(s.size() > t.size())
        t = new LinkedList<Integer>(s);
    return (is_sorted)? L.iterator() : t.iterator() ;
}
```

F(n) = n => F(n) = Θ(n) ; n = size of the input linkedlist

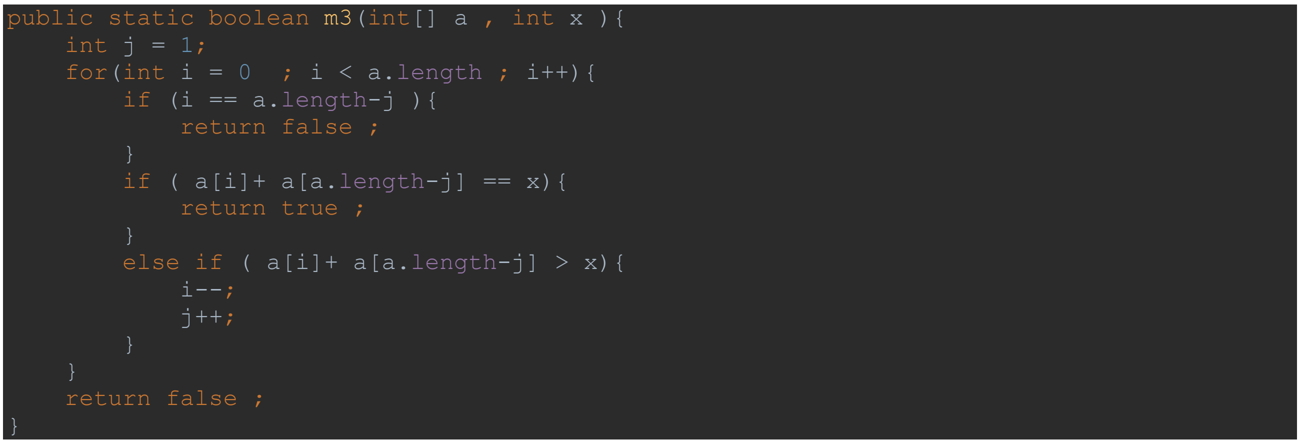
b)

```
public static LinkedList<Integer> longest_sorted_re(Iterator<Integer> in ,
LinkedList<Integer> out ,LinkedList<Integer> temp ){
    if(in.hasNext()){
        int j = in.next();
        if( !out.isEmpty() && j < out.getLast()){
            if(out.size() > temp.size() ){
                temp = new LinkedList<Integer>(out);
            }
            out.clear();
        }
        out.add(j);
        return longest_sorted_re(in,out ,temp);
    }
    else {
        return temp;
    }
}
```

T(n) = T(n - 1) +1 =>

a) Master theorem :
T(n) = aT(n – b) + F(n) ; F(n) = O(n^k)
In this case
a = 1 ;
b = 1 ;
F(n) = 1 => k = 0 ;
So according to master theorem
T(n) = O(n^(k +1)) =>
T(n) = O(n) ;

2.



```
public static boolean m3(int[] a , int x ){
    int j = 1;
    for(int i = 0  ; i < a.length ; i++){
        if (i == a.length-j ){
            return false ;
        }
        if ( a[i]+ a[a.length-j] == x){
            return true ;
        }
        else if ( a[i]+ a[a.length-j] > x){
            i--;
            j++;
        }
    }
    return false ;
}
```

F(n) = 7n + 2

To say that F(n) = Θ(n) we have to find to contant C1 and C2 that

C1*n ≤ F(n) ≤ C2*n

C1*n ≤ 7n + 2 ≤ C2*n ;

For C1 = 1 ; C2 = 9 ; n = 1 the equation is true so

F(n) = Θ(n) ;

3.

$$\mathrm{f(n)} = \sum_{i=1}^{2n} (\textstyle\sum_{j=1}^{i} \frac{j}{3} + 1)$$

$$\frac{i*(i+1)}{6} + 6$$

$$\frac{1}{6}\textstyle\sum_{i=1}^{2n}(i^2 + 7i)$$

4.

$$T(n) = 4T\left({}^{n}/_{2}\right) + \frac{n^2}{4}$$

According to master theorem

$T(n) = aT\left({}^{n}/_{b}\right) + F(n)$

$F(n) = \ \Theta(n^k\ (\log n)^p)$

In this case

a = 4 ;

b = 2 ;

k = 2 ;

p = 0 ;

$\log_2 4 = 2$

P > -1 ; =>

$T(n) = \Theta(n^k (\log n)^{p+1}\ )$ =>

$T(n) = \Theta(\ n^2 \log n\ \ )$