

**Gebze Technical University
Computer Engineering**

CSE 222 - 2019 Spring

HOMEWORK 4 REPORT

**Galip Tayfun Saygılı
161044083**

Course Assistant: Erchan Aptoula

1 System Requirements

Hardware requirements

- 2 GB RAM minimum, 4 GB RAM recommended
- 1.5 GB hard disk space + at least 1 GB for caches
- 1024x768 minimum screen resolution

Software requirements

- JRE 1.8 is bundled with the IntelliJ IDEA distribution. You do not need to install Java on your computer to run IntelliJ IDEA.
- A standalone JDK is required for Java development.

Linux

- OS Linux (note that a 32-bit JDK is not bundled, so a 64-bit system is recommended)
- KDE, Gnome or Unity desktop environment recommended

2 Solutions

2.1 Question 1

```
public static LinkedList findlargest(LinkedList list){
    if(list == null){
        return new LinkedList();
    }
    else {
        Node temp = new Node();
        temp = list.head();
        int max = 0;
        int tempmax;
        int index = -1;
        int tempindex = 0;
        while (temp != null) {
            Node temp2 = new Node();
            temp2 = temp;
            tempmax = 1;
            while (temp2.next() != null) {
                if(temp2.node() <= temp2.next.node()){
                    temp2 = temp2.next();
                    ++tempmax;
                }
                else break;
            }
            if (tempmax > max){
                max = tempmax;
                index = tempindex;
            }
            ++tempindex;
            temp = temp.next();
        }
    }
}
```

```

        temp = list.head();
        for (int i=0; i<index; ++,)
            temp = temp.next();
        LinkedList templist = new LinkedList();
        templist.setHead(new Node());
        Node traverse = templist.head();
        traverse.setNode(temp.node());
        for (int i=1; i<max;++i){
            traverse.setNext(new Node());
            traverse = traverse.next();
            temp = temp.next();
            traverse.setNode(temp.node());
        }
        return templist;
    }
}

```

$$O((n*(n+1))/2) = O(n^2)$$

```

public static LinkedList findlargestrecursive(LinkedList list){
    if(list == null){
        LinkedList temp = new LinkedList();
        temp.setLength(0);
        return temp;
    }
    else {
        Node temp = new Node();
        temp = list.head();
        int max = 1;
        while (temp.next() != null) {
            if(temp2.node() <= temp2.next.node()){
                temp2 = temp2.next();
                ++max;
            }
            else break;
            temp = temp.next();
        }
        temp = list.head();
        LinkedList temprecursive = new LinkedList();
        temprecursive.setHead(temp.next());
        LinkedList templist2 = findlargestrecursive(temprecursive);
        LinkedList templist = new LinkedList();
        templist.setHead(new Node());
        templist.setLength(max);
        Node traverse = templist.head();
        traverse.setNode(temp.node());
        for (int i=1; i<max;++i){
            traverse.setNext(new Node());
            traverse = traverse.next();
            temp = temp.next();
            traverse.setNode(temp.node());
        }
        if(templist2.getLength() > templist.getLength())
            return templist2;
        else return templist;
    }
}

```

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

$$T(n) = n + (n-1) + (n-2) + \dots + 2 + 1 + T(0)$$

$$T(0) = 1$$

$$O(((n*(n+1))/2)+1) = O(n^2/2 + n/2 + 1) = O(n^2)$$

2.2 Question 2

```
public static int[] findsum(int[] arr, int x){
    int i = 0;
    int j = arr.length - 1;
    while(i < j){
        if(arr[i] + arr[j] == x){
            int[] temp = new int[2];
            temp[0] = arr[i];
            temp[1] = arr[j];
            return temp;
        }
        else if(arr[i] + arr[j] > x){
            --j;
        }
        else{
            ++i;
        }
    }
    int[] noexist = new int[2];
    noexist[0] = -1;
    noexist[1] = -1;
}
```

2.3 Question 3

2.4 Question 4

$$T(n) = 4 \cdot T(n/2) + (n/2) \cdot (n/2) \cdot 4 + 5 \quad (T(1) = 1)$$

$$T(n) = 4 \cdot T(n/2) + 2 \cdot n^2 + 5$$

By using master theorem;

Since $a \geq 1$ ($a = 4$) and $b > 1$ ($b = 2$), the time complexity is:

$\log_b a = 2$ then,

$$T(n) = O(n^2 \cdot \log(n))$$

2.5 Question 5

In the hw4 archive folder.