

SEN2211 Data Structures and Algorithms I

Sample Questions

Question 1

- **(10 pts)** Determine the complexity of the following implementations. Explain your answer.

```
for(int j = 1; j <= n / 2; j++)  
    for (int k = 1; k <= n * n; k++)  
        x = x + j + k;
```

Outer loop executes $n/2$ times. For each of those times, inner loop executes n^2 times, so the body of the inner loop is executed $(n/2) * n^2 = n^3/2$ times. The algorithm is $O(n^3)$.

Question 2

- **(10 pts)** Determine the complexity of the following implementations. Explain your answer.

```
for(int j = 1; j <= n; j++)  
    for(int k = 1; k < 3 * j; k++)  
        x = x + j;
```

When j is 1, inner loop executes 3 times; when j is 2, inner loop executes $3 \cdot 2$ times; ... when j is n , inner loop executes $3 \cdot n$ times. In all the inner loop executes $3+6+9+\dots+3n = 3(1+2+3+\dots+n) = \frac{3n^2}{2} + \frac{3n}{2}$ times. The algorithm is $O(n^2)$.

Question 3

- **(20 pts)** Assume that you have a Stack implemented with an array. What is the output of the following code if the user enters the following input in the given order:
- 24 15 14 33 10 5 6 4 12 -999

Question 3 cont.

```
public static void main(String[] args)
{
    Stack a = new Stack(10);
    int x, i = 2;
    Scanner s = new Scanner(System.in);
    a.push(5);
    x = s.nextInt();
    while (x != -999) {
        if (x * i % 3 == 0){
            if(!a.isFull())
                a.push(x);

            i++;
        }
        else if (x * i % 3 == 1){
            a.pop();
            i--;
        }
        else
            System.out.println("x = " + x);
        x = s.nextInt();
    }
}
```

```
System.out.println("Stack Elements:");
while (!a.isEmpty())
{
    System.out.println(a.top());
    a.pop();
}
```

24
15
14
x = 14
33
10
x = 10
5
6
4
x = 4
12
-999
Stack
Elements:
12
6
15
24
5

Question 4

- You are given the following class definition for a linked list node and a diagram for the corresponding ordered linked list:

```
public class Node {  
    private int info;  
    private Node link;  
}
```

```
public class OrderedLinkedList {  
    private Node first;  
    private Node last;  
}
```



Question 4 cont.

- **(5 pts)** Write Java statements that adds info 68 to this ordered linked list. (No search operation!!!)

```
Node x = new Node(68, null);  
list.getLast().setLink(x);  
list.setLast(x);
```

Question 4 cont.

- **(10 pts)** Write the Java statements that adds info 40 to this ordered linked list. You must find the right place in the linked list for this insertion. You know from the figure that the item will be inserted somewhere in the list.

```
boolean f = false;
Node current = new Node();
Node prev = new Node();
Node a = new Node();
a.setInfo(40);
current = list.getFirst();
while(current != null && !f){
    if(current.getInfo() > a.getInfo())
        f = true;
    else{
        prev = current;
        current = current.getLink();
    }
}
prev.setLink(a);
a.setLink(current);
```


Question 5

- **(25 pts)** Write a method called `reverseList()` reverses the order of the elements in a linked list. The method `reverse` takes no reference and returns nothing. It will be called by a linked list object and reverse the order of the elements of it. (do not use arrays in your solution, solve it with linked lists!)

Question 5 cont.

```
public void reverseList() {  
    // curNode traverses the list, first is reset to empty list.  
    Node curNode = first;  
    Node nxtNode = new Node();  
  
    first = null;  
  
    // Until no more in list, insert current before first and  
    //advance.  
    while (curNode != null) {  
        // Need to save next node since we're changing the  
        //current.  
        nxtNode = curNode.getLink();  
  
        // Insert at start of new list.  
        curNode.setLink(first);  
        first = curNode;  
        // Advance to next.  
        curNode = nxtNode;  
    }  
}
```

Question 6

- Answer the following questions.
 - **(10 pts)** Find the value of the postfix expression given below. Show the stack operations.
 - $3\ 4\ *\ 7\ 2\ 11\ /\ -\ 5\ +\ /\ =\ ?$
 - **(15 pts)** Write the infix expression given below in postfix form.
 - $(a + b) * c / (d - e + f * g) - (h + i)$