Prince Sultan University
CCIS - Department of Computer Science

Major Exam 1
Term 231

Course title: Data Structures and Algorithms	**Course Code**: CS 210
Exam date: 2/10/2023	**Exam Time: 50** minutes

Student Name: ____________________________

Student ID:									

Section: Circle your instructor name and the class time.

Instructor	**Dr. Syed Umar Amin**		**Dr. Basit Qureshi**	**Dr. Abdullah Alrajeh**
Time	8	9	10	11

Question Number	CLO	Question points	Score
Question 1	**CLO 1**	5	
Question 2	**CLO 2**	7	
Question 3	**CLO 4**	3	
Total out of		**15**	

Instructions:

- This exam contains three questions with multiple parts.
- Time allowed: 50 minutes
- Closed Book, Closed Notes.
- Use of Calculators and / or computing devices / smartphones etc is strictly prohibited.
- Answer the problems on the exam sheets only. No additional attachments would be accepted.
- If you need extra space use the back of a page.
- When the “time is over” is called, it is students’ responsibility to submit his exam to the invigilator. Submitting completed exam 3 minutes after the “time is over” will incur a penalty of **5 points**.
- Do **NOT** use the erasable pens

Few gentle reminders:

- If you get stuck on some problem for a long time, move on to the next one.
- The ordering of the problems is somewhat related to their relative difficulty. However, the order might be different for you!
- You should be better off by first reading all questions and answering them in the order of what you think is the easiest to the hardest problem.
- Keep the points distribution in mind when deciding how much time to spend on each problem.

Question 1 [5 points - CLO 1]

Part A: **(4 / points)**

Consider a Node class that contains an `integer val` and `Node next` as attributes.

1. **Write a Java method `PrintDuplicate`() extending the SinglyLinkedList class. This method prints all duplicate values in the List.**
2. **Explain why or why not your code runs in linear time.**

Part B: (1 / points)

What does the following code do?

```
public Node Find(SinglyLinkedList L) {
    if (L.head == null) {
        return null;
    }
    Node slow = L.head;
    Node fast = L.head;
        while (fast != null && fast.next != null) {
        slow = slow.next;
        fast = fast.next.next;
    }
    return slow;
}
```

A. Find the middle element in a singly linked list
B. Find the second to last element in a singly linked list
C. Find the last element in a singly linked list
D. None of the above

Question 2 [7 points - CLO 2]

Part A: (5 / points)

Find the estimate for T(n) and the Asymptotic complexity (Big O) for the following algorithms by computing the number of primitive operations:

Code	Estimate T(n)	O(n)

`int k= 0;`
`for (i = 1; i <= n; i++) {`
`   for (j = 1; j <= 10; j++) {`
`      k = k + i + j;`
`   }`
`}`		

`int k= 0;`
`for (i = 1; i <= n; i++) {`
`   for (j = i; j <= n; j++) {`
`     k = k + i`
`    }`
`}`		

`for (i = n; i <= 1; i/=2) {`
` System.out.println(i);`
`}`		

<table>
<tr><th>Code</th><th>Estimate T(n)</th><th>O(n)</th></tr>
<tr><td><pre><code>public void print(int n){
 int k = n;
 for (int i = 1; i <= n; i++)
    for (int j = 1; j <= k; j++)
       System.out.println(j+i);
}
public void main(){
  Scanner Key= new Scanner(System.in);
  int n=Key.nextInt();
  for (i = 1; i <= n; i++)
    print(n);
}}</code></pre></td><td></td><td></td></tr>
<tr><td><pre><code>void f1(int n) {
  if (n==1)
    return;
  for (int i=1; i<=n; i++) {
        for (int j=1; j<=n; j++) {
            System.out.println("*");
            break;
        }
  }
}</code></pre></td><td></td><td></td></tr>
</table>

Part B: (2 / points)

For each function f(n) below, give an asymptotic upper bound using “big-Oh” notation.

(a) $f(n) = 100n^2 - 100n^2 + 14\,n^3$ ___________

(d) $f(n) = 100 \log\log n + 100 \log^2 n$ ___________

(f) $f(n) = n^3\,(10 + 20n + 20n^2)$ ___________

(h) $f(n) = n^2 \log n + 100n$ ___________

Question 3 [3 points - CLO 4]

Part A: **(2 / points)**

1. **Give the running time T(n) of the following function and provide the Big-Oh notation.**
2. **Show the output when n=3 and draw the recursion trace.**

```
static void fun(int n)
{
    if (n < 1)
        return;
    else {
        System.out.println( n);
        fun(n - 1);
        System.out.println(, n);
        return;
    }
}
```

Part B: **(1 / point)**

Algorithms A and B spend exactly $T_A(n) = 5 \cdot n \cdot \log_2 n$ and $T_B(n) = 25 \cdot n$ microseconds, respectively, for a problem of size n. Which algorithm is better in terms of time complexity? For which problem size (c and n_0) does it outperform the other?

******************************* End of Exam *******************************