Prince Sultan University
CCIS - Department of Computer Science

Major Exam 1
Term 242

Course Title: Data Structures and Algorithms **Course Code**: CS 210

Exam date: 03/02/2025 **Exam Time: 50** minutes

Student Name:

Student ID:

Section #:

Serial #:

Question Number	CLO	Question points	Score
Question 1	**CLO 1**	7	
Question 2	**CLO 3**	3	
Question 3	**CLO 4**	5	
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.

Q1 – Part A	**1.**	Ⓐ Ⓑ Ⓒ Ⓓ
	2.	Ⓐ Ⓑ Ⓒ Ⓓ
	3.	Ⓐ Ⓑ Ⓒ Ⓓ
	4.	Ⓐ Ⓑ Ⓒ Ⓓ
	5.	Ⓐ Ⓑ Ⓒ Ⓓ
	6.	Ⓐ Ⓑ Ⓒ Ⓓ
	7.	Ⓐ Ⓑ Ⓒ Ⓓ
	8.	Ⓐ Ⓑ Ⓒ Ⓓ
Q1 – Part B	**9.**	Ⓐ Ⓑ Ⓒ Ⓓ
	10.	Ⓐ Ⓑ Ⓒ Ⓓ
	11.	Ⓐ Ⓑ Ⓒ Ⓓ
Q2 – Part A	**12.**	Ⓐ Ⓑ Ⓒ Ⓓ
	13.	Ⓐ Ⓑ Ⓒ Ⓓ
	14.	Ⓐ Ⓑ Ⓒ Ⓓ
Q3 – Part C	**15.**	Ⓐ Ⓑ Ⓒ Ⓓ
	16.	Ⓐ Ⓑ Ⓒ Ⓓ
	17.	Ⓐ Ⓑ Ⓒ Ⓓ
	18.	Ⓐ Ⓑ Ⓒ Ⓓ

Question 1 [/ 6 points - CLO 1]

Part A: (/ 4 points)

1. What is the time complexity for inserting a new node at the beginning of a doubly linked list?

a) O(1)	b) O(n)	c) O(log n)	d) O(n^2)

2. What does the following code do? While traversing, we are at a "current" node in the linked list.

```
Node temp = new Node();
temp.element = 10;
temp.prev = current;
temp.next = current.next;
current.next.prev = temp;
current.next = temp;
```

a) Deletes a node in a doubly linked list.	b) Insert a new node in a doubly linked list.	c) Traverses the list backward.	d) Reverses the list

3. What will the following operation's output be if the list contains nodes with elements 1→ 2→3→4, and the current node is 2?

```
current.prev.next = current.next;
current.next.prev = current.prev;
```

a) List will contain 1, 2, 4.	b) List will contain 1, 3, 4.	c) The list will remain unchanged.	d) The list will throw an exception.

4. What is the time complexity of deleting a node at the tail of a singly linked list without using a tail pointer?

a) O(1)	b) O(n)	c) O(log n)	d) O(n^2)

5. What is the result of the following code for a singly linked list with elements 1 → 2 → 3?

```
Node current = head;
while (current != null && current.data != 2) {
    current = current.next;
}
if (current != null) {
    current.data = 5;
}
```

a) Updates the head to 5.	b) Updates the tail to 5.	c) Replace the value 2 with 5.	d) Throws a null pointer exception.

6. What does the following code do in a circular linked list?

```
Node temp = head;
while (temp.next != head) {
    temp = temp.next;
}
temp.next = new Node(10);
temp.next.next = head;
```

a) Insert a new node at the beginning.	b) Insert a new node at the end.	c) Delete the last node.	d) Traverse the list.

7. What does the following code to a single linked list?

```
Node prev = null;
Node current = head;
while (current != null) {
    Node next = current.next;
    current.next = prev;
    prev = current;
    current = next;
}
head = prev;
```

a) Traverses the list but does not reverse it.	b) Reverses the singly linked list.	c) Deletes the entire list.	d) Throws an exception.

8. What does the following code do in a circular linked list?

```
if (head != null && head.next == head) {
    head = null;
} else {
    Node temp = head;
    while (temp.next.next != head) {
        temp = temp.next;
    }
    temp.next = head;
}
```

a) Delete the last node.	b) Delete the first node.	c) Insert a node at the end.	d) Traverses the list.

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

The following Implements a method `insertInOrder` in the class DoublyLinkedList that accepts an integer X as an argument. This method will create a new node with key X and insert it into the list in the appropriate position to maintain the list's order.

```
 1  class DoublyLinkedList {
 2   class Node {
 3   int key;
 4   Node prev, next;
 5
 6   public Node(int key) {
 7   this.key = key;
 8   this.prev = this.next = null;
 9   }
10   }
11   public void insertInOrder(Node head, int x) {
12   Node newNode = new Node(x);
13
14   if (_______________) {
15   newNode.next = head;
16   head.prev = newNode;
17   head = newNode;
18   return;
19   }
20
21   Node curr = head;
22   while (___________________________) {
23   curr = curr.next;
24   }
25
26   newNode.next = curr.next;
27   if (_________________) {
28   curr.next.prev = newNode;
29   }
30   curr.next = newNode;
31   newNode.prev = curr;
32   return;
33
     } }
     //other methods for remove/inserts/search etc.
```

9. What is an appropriate statement for the if condition on line 14.

A) `x <= head.data`	B) `head.data <= x`
C) `head.data == X`	D) `head.next.data <= X`

10. What is an appropriate statement for the while loop condition on line 22

A) `curr.next != null && curr.data < x`	B) `curr != null && curr.data < x`
C) `curr.next != null && curr.next.data < x`	D) `curr != null`

11. What is an appropriate statement for the if condition on line 27.

A) `curr.next.next != null`	B) `curr.next != null`
C) `curr == null`	D) `curr != null`

Question 2 [/ 3 points - CLO 3]

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

Answer the following MCQs

12. How many primitive operations are executed by the following code segment?

```
int sum = 0;
for (int i = 0; i < N; i++) {
    sum += i;
}
```

A) N	B) N^2+3	C) 4N+3	D) 3N - 2

13. What is the Big O complexity of the expression $5N^2+3N+10$?

A) O(N)	B) $O(N^2)$	C) $O(N^3)$	D) O(log N)

14. For the nested loop below, what is the time complexity, for n >= 1?

```
for( int i = n;  i > 0;  i /= 2 ) {
  for( int j = 1;  j < n;  j *= 2 ) {
    for( int k = 0;  k < n;  k += 2 ) {
          ... // constant number of operations
    }
  }
}
```

A) O(log N)	B) $O(N^2)$	C) $O(N^3)$	D) O(n(log n)^2)

Part B: **(/ 1.5 point)**

Algorithms **A** and **B** spend exactly $T_A(n) = 3n^2 + 10n$ and $T_B(n) = 20n \log n$, respectively, for a problem of size n. Find out the value of n_0 where algorithm B is better than algorithm A.

n0=32

Question 3 [/5 points - CLO 3]

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

Consider the following recursive function that counts the number of `1`s in the binary representation of a number:

```
int count_ones(int n)
{
        if (n == 0)
        return 0
        else
                return (n % 2) + count_ones(n / 2)
}
```

1. Trace the recursive calls and their return values for count_ones(13). [1.5 points]

Recursion Trace for `count_ones(13)`:

1. `count_ones(13)`
   - `n = 13`, so `13 % 2 = 1` (since 13 is odd, the least significant bit is 1)
   - Calls `count_ones(13 // 2)` which is `count_ones(6)`
2. `count_ones(6)`
   - `n = 6`, so `6 % 2 = 0` (since 6 is even, the least significant bit is 0)
   - Calls `count_ones(6 // 2)` which is `count_ones(3)`
3. `count_ones(3)`
   - `n = 3`, so `3 % 2 = 1` (since 3 is odd, the least significant bit is 1)
   - Calls `count_ones(3 // 2)` which is `count_ones(1)`
4. `count_ones(1)`
   - `n = 1`, so `1 % 2 = 1` (since 1 is odd, the least significant bit is 1)
   - Calls `count_ones(1 // 2)` which is `count_ones(0)`
5. `count_ones(0)`
   - Base case: `n = 0`, returns 0

`count_ones(0)` returns 0.

`count_ones(1)` returns `1 + 0 = 1`.
`count_ones(3)` returns `1 + 1 = 2`.
`count_ones(6)` returns `0 + 2 = 2`.
`count_ones(13)` returns `1 + 2 = 3`.

2. Find the Big-O notation for the method above. [0.5 point]

Thus, the final result is 3. The Big O is O(logn).

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

What is the time complexity of the following recursive functions?

15.
```
public static int recursiveFunction(int n) {
 if (n <= 1) {
        return 1;
 }
else
        return recursiveFunction(n - 1) + recursiveFunction(n - 1);
 }
```

A) $O(\log_2 n)$	B) $O(n)$	C) $O(n^2)$	D) $O(2^n)$

16.
```
public static void traverseList(List<?> list, int index) {
 if (index == list.size()) {
        return;
 }
        System.out.println(list.get(index));
        traverseList(list, index + 1);
 }
```

A) $O(\log_2 n)$	B) $O(n)$	C) $O(n^2)$	D) $O(2^n)$

17.
```
public static int sumBinary(int n) {
        if (n == 0) {
                return 0;
        }
        else
                return (n % 2) + sumBinary(n / 2);
 }
```

A) $O(\log_2 n)$	B) $O(n)$	C) $O(n^2)$	D) $O(2^n)$

18.
```
public static void printPairs(int[] array, int i, int j) {
 if (i == array.length) {
        return;
 }
 if (j < array.length) {
        System.out.println("Pair: (" + array[i] + ", " + array[j] + ")");
        printPairs(array, i, j + 1);
        }
else {
        printPairs(array, i + 1, i + 1);
        }
 }
```

A) $O(\log_2 n)$	B) $O(n)$	C) $O(n^2)$	D) $O(2^n)$

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