Prince Sultan University
CCIS - Department of Computer Science

Mid Exam
Term 251

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

Exam date: 29/09/2025 **Exam Time: 50 minutes**

Student Name:

Student ID:

Section #:

Serial #:

Question Number	CLO	Question points	Score
Question 1	**CLO 1**	6	
Question 2	**CLO 3**	10	
Question 3	**CLO 4**	4	
Total out of		**20**	

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.

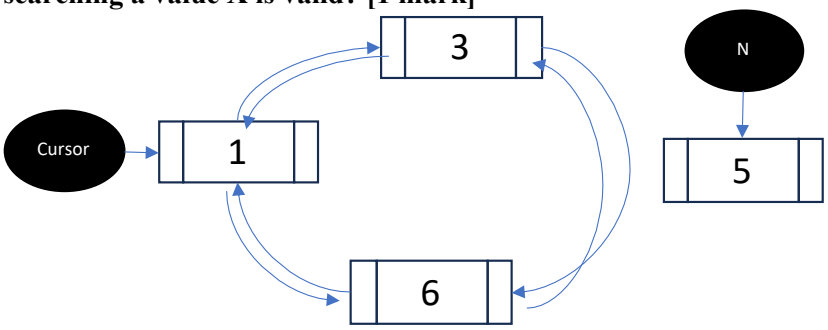
Prince Sultan University
CCIS - Department of Computer Science

Note: Only answers marked on this page would be graded.

Q1	**1.**	Ⓐ Ⓑ Ⓒ Ⓓ	**2.**	Ⓐ Ⓑ Ⓒ Ⓓ
	3.	Ⓐ Ⓑ Ⓒ Ⓓ	**4.**	Ⓐ Ⓑ Ⓒ Ⓓ
	5.	Ⓐ Ⓑ Ⓒ Ⓓ	**6.**	Ⓐ Ⓑ Ⓒ Ⓓ
	7.	Ⓐ Ⓑ Ⓒ Ⓓ	**8.**	Ⓐ Ⓑ Ⓒ Ⓓ
	9.	Ⓐ Ⓑ Ⓒ Ⓓ		
Q2 – Part A	**10.**	Ⓐ Ⓑ Ⓒ Ⓓ	**11.**	Ⓐ Ⓑ Ⓒ Ⓓ
	12.	Ⓐ Ⓑ Ⓒ Ⓓ	**13.**	Ⓐ Ⓑ Ⓒ Ⓓ
	14.	Ⓐ Ⓑ Ⓒ Ⓓ	**15.**	Ⓐ Ⓑ Ⓒ Ⓓ
	16.	Ⓐ Ⓑ Ⓒ Ⓓ	**17.**	Ⓐ Ⓑ Ⓒ Ⓓ
	18.	Ⓐ Ⓑ Ⓒ Ⓓ		
Q2 – Part B	**19.**	Ⓐ Ⓑ Ⓒ Ⓓ	**20.**	Ⓐ Ⓑ Ⓒ Ⓓ
Q3	**21.**	Ⓐ Ⓑ Ⓒ Ⓓ	**22.**	Ⓐ Ⓑ Ⓒ Ⓓ
	23.	Ⓐ Ⓑ Ⓒ Ⓓ	**24.**	Ⓐ Ⓑ Ⓒ Ⓓ
	25.	Ⓐ Ⓑ Ⓒ Ⓓ	**26.**	Ⓐ Ⓑ Ⓒ Ⓓ
	27.	Ⓐ Ⓑ Ⓒ Ⓓ	**28.**	Ⓐ Ⓑ Ⓒ Ⓓ

Question 1 [/ 6 points - CLO 1]

1. In figure 1, a circular list is given that traverses in both directions. Which "while" code snippet for searching a value X is valid? [1 mark]



A)
```
while(Cursor.next!=null){
 if(Cursor.value==X) return Cursor;
 Cursor=Cursor.next;}
```

B)
```
while(Cursor.prev!=null){
 if(Cursor.value==X) return Cursor;
 Cursor=Cursor.prev;}
```

C)
```
for(int i=1;i<=size;i++){
 if(Cursor.value==X) return Cursor;
 Cursor=Cursor.next;}
```

D)
```
for(int i=0;i<=size;i++){
 if(Cursor.value==X) return Cursor;
 Cursor=Cursor.next;}
```

2. As shown in figure 1, which of the following would insert a node N after Node containing the element 3? [1 mark]

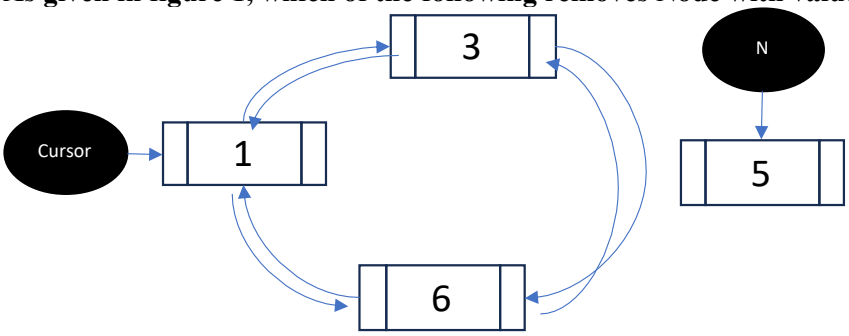
A) `N.next = Cursor; Head.next = Cursor; Head = Cursor; Cursor = Cursor.prev; N.next.prev = N;`

B) `Cursor = N; Cursor.next = N.next; N.next = Cursor; Cursor = Cursor.next; N.next.prev = N;`

C) `N.next = Cursor.next; Cursor.prev = N; Head = N; Cursor = Cursor.next; Cursor.next = N;`

D) `Cursor = Cursor.next; N.prev = Cursor; N.next = Cursor.next; Cursor.next = N; N.next.prev = N;`

3. As given in figure 1, which of the following removes Node with value 1? [1 mark]




```
A) Cursor = null;
B) Cursor.next.prev = Cursor.prev; Cursor.prev.next = Cursor.next; Cursor=Cursor.next;
C) Cursor.next.prev = Cursor.next; Cursor.prev.next = Cursor.prev; Cursor=Cursor.next;
D) Cursor=Cursor.next; Cursor.prev = Cursor.prev; Cursor.next = Cursor.next;
```

4. What is the time-complexity of inserting a node before the head in an un-ordered circular linked list? [0.5 mark]

A) O(1)
B) O(n)
C) O(nlogn)
D) $O(n^2)$

5. In a doubly linked list, how many next and prev pointers are null? [0.5 mark]

A) None.
B) One next and one prev.
C) Two next pointers.
D) One prev pointer.

6. Consider the following two code fragments, if both lists contain the same number of nodes (n > 0), which of the following statements is correct? [0.5 mark]

Code A (Singly Linked List traversal)	Code B (Circular Linked List traversal)

`Node current = head;`
`while (current != null) {`
`  System.out.print(current.data + " ");`
`  current = current.next;`
`}`
	`Node current = head; // Assume head is defined`
	`while (current.next != head) {`
	`  System.out.print(current.data + " ");`
	`  current = current.next;`
	`}`

A) Code A will print all nodes once, Code B will print all nodes once.	B) Code A will print all nodes once, Code B will print only n-1 nodes.	C) Code A will run infinitely, Code B will terminate correctly after printing all nodes.	D) Both Code A and Code B will run infinitely.

7. What is wrong with the following code for inserting a new node **after cursor in a** doubly circular linked list**? [0.5 mark]**

```
Line1: newNode.next = cursor.next;
Line2: newNode.prev = cursor;
Line3: cursor.next = newNode;
Line4: newNode.next.prev = newNode.next;
```

A) Line 1 is wrong: `newNode.next` should point to `cursor`.	B) Line 2 is wrong: `newNode.prev` should point to `cursor.next`.	C) Line 4 is wrong: `newNode.next.prev` should be assigned to `newNode`, not `newNode.next`.	D) There is nothing wrong; the code correctly inserts the node.

8. If you want to insert a node in the middle **(at position n/2) of a singly linked list, what is the time complexity? [0.5 mark]**

A) O(1)	B) O(n)	C) O(log n)	D) $O(n^2)$
9. Which of the following operations has different running time **between a singly and a doubly linked list? [0.5 mark]**			
A) Insertion at the head	B) Searching by value	C) Deletion of a given node (if pointer to node is provided)	D)Traversal of the entire list

Question 2 [/ 10 points - CLO 3]

Part A: **(/ 6 points)**

Answer the following MCQs

10. You need to delete an element from a 1D unsorted array at a specific index i, the placement of the remaining elements does not change after deletion. What is the worst-case time complexity for this operation? [0.5 mark]			
A) O(1)	B) O(logn)	C) O(n)	D) O(n^2)
11. What is the time complexity of reversing only the first half of a doubly linked list of n elements (where n is even)? [0.5 mark]			
A) O(1)	B) O(logn)	C) O(n)	D) O(n^2)
12. You are given a value v that exists in a doubly linked list with no duplicate values. What is the best-case time complexity to find and delete the node containing v? [0.5 mark]			
A) O(1)	B) O(logn)	C) O(n)	D) O(n^2)
13. In a singly linked list of integers with both head and tail pointers, what is the time complexity of finding the median element? [0.5 mark]			
A) O(1)	B) O(logn)	C) O(n)	D) O(n^2)
14. Which statement best compares the time complexity of prepending an element (adding to the front)? [0.5 mark]			
A) Array: O(1), Doubly Linked List: O(n)	B) Array: O(n), Doubly Linked List: O(1)	C) Both are O(1)	D) Both are O(n)

15. What would be the time complexity of the following functions? [0.5 mark] $\mathbf{f(n) = 2^n + n^5 + 1000n^{1/2}}$
a. $O(n^4)$
b. $O(n^{1/2})$
c. $O(n^5)$
d. $O(2^n)$

16. Consider two methods, A and B, with the following time complexity:
A: 2^n
B: n^3
Which of the following statements best describes their growth? [1 mark]

a. A's execution time remains smaller than B's for every value of n.
b. A is more efficient for small inputs but eventually grows faster than B as n increases.
c. Both methods scale identically in the asymptotic sense.
d. B's execution time remains smaller than A's for every value of n.

17. What would be the time complexity of the following code? [1 mark]

```
void method(int n){
        int i = 1;
        int j = 1;
  while (i <= n) {
    while (j <= n) {
      j = j + 1;
    }
    i = i * 2;
  }
}
```

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

18. What would be the time complexity of the following code? [1 mark]

```
void method(int n) {
        for(int i = 0; i < n; i++) {
                for(int j = 0; j < n; j++) {
                        for(int z = 0; z < n*n; z++) {
                        }
                }
        }
}
```

a. $O(n^3)$
b. $O(n^2)$
c. $O(n^2 \log n)$
d. $O(n^4)$

Part B: (/ 2 points)

What is the time complexity of the following recursive functions? [1 mark each]

19.

```
int g(int n) {
  if (n == 0) return 1;
  return g(n - 1) + g(n - 1) + g(n - 1);
}
```

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

```
20.    public static int sumDigits(int n) {
         if (n == 0) {
           return 0;
         } else {
            return (n % 10) + sumDigits(n / 10);
         }
       }
```

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

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

Consider the following recursive function:

```
public static int paths(int m, int n) {

    if (m == 0 || n == 0) return 1;

    return paths(m - 1, n) + paths(m, n - 1);
}
```

1. Trace the recursive calls and their return values for **`paths(1,3)`** [1+ 0.5 points]
2. Find the Big-O notation for the method above. [0.5 point]

Question 3 [/4 points - CLO 3]

Consider the following code for a sorting algorithm:

```
1 int[] A = {10, 3, 8, 15, 6};
2  for (int i = 0; i < A.length - 1; i++) {
3    int min = i;
4    for (int j = i + 1; j < A.length; j++)
{
5         if (A[j] < A[min]) min = j;
6    }
7    if (min != i) {
8       int temp = A[i];
9       A[i] = A[min];
10     A[min] = temp;
11   }
12    System.out.print(A[2] + " ");
13 }
```

21. What is the state of the array after iteration i = 2? [0.5 mark]

a) {3, 6, 8, 15, 10} with a swap performed
b) {3, 6, 8, 15, 10} with no swap performed
c) {3, 6, 10, 15, 8} with a swap performed
d) {3, 6, 10, 15, 8} with no swap performed

22. When the array is already sorted, how many comparisons does the above sort make? [0.5 mark]

a) O(n)
b) O(log n)
c) ($O(n^2)$)
d) None

23. When the array is already sorted, how many swaps does the above sort make? [0.5 mark]

a) n−1
b) n
c) 0
d) n(n+1)/2

Consider the Selection Sort Algorithm. Two students make the following claims:

- **Student A:** *"Selection Sort always performs the same number of comparisons, regardless of the input order."*
- **Student B:** *"Selection Sort also always performs the same number of swaps, regardless of the input order."*

24. Which student's statement is correct? [0.5 mark]

a) Only Student A is correct
b) Only Student B is correct
c) Both A and B are correct
d) Neither A nor B is correct

Consider the following code for insertion sort algorithm:

```
int[] A = {12, 4, 7, 9, 1};
for (int i = 1; i < A.length; i++) {
    int key = A[i];
    int j = i - 1;
    while (j >= 0 && A[j] > key) {
        A[j + 1] = A[j];
        j = j - 1;
    }
    A[j + 1] = key;
}
```

25. In iteration i = 3, which comparison(s) are made? [0.5 mark]

a) 12 vs 9
b) 7 vs 9
c) 4 vs 9
d) Both (a) and (b)

26. What is the array state after iteration i = 3 (key = 9) ? [0.5 mark]

a) {4, 7, 9, 12, 1}
b) {4, 7, 12, 9, 1}
c) {12, 4, 7, 9, 1}
d) {4, 9, 7, 12, 1}

27. During iteration i = 3 (key = 9), how many shifts (swaps) occur? [0.5 mark]

a) 1
b) 2
c) 3
d) 4

28. What is the worst-case time complexity (comparisons) of Insertion Sort when the array is in reverse order? [0.5 mark]

a) O(n)
b) O(log n)
c) O(nlog n)
d) O(n2)

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