



# UNIVERSITY OF THE PUNJAB

B.S. 4 Years Program / Fourth Semester – 2019

Paper: Data Structure and Algorithm

Course Code: IT-207 / IT-22408 Part – I (Compulsory)

Time: 15 Min. Marks: 10

Roll No. in Fig. ....

Roll No. in Words. ....

Signature of Supdt.:

**ATTEMPT THIS PAPER ON THIS QUESTION SHEET ONLY.**

**Division of marks is given in front of each question.**

**This Paper will be collected back after expiry of time limit mentioned above.**

**Q.1. Encircle the right answer cutting and overwriting is not allowed. (10x1=10)**

1. What is the worst case time complexity of linear search algorithm?

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

2. After each iteration in bubble sort

- a) at least one element is at its sorted position.
- b) one less comparison is made in the next iteration.
- c) Both A & B are true.
- d) Neither A or B are true.

3. Heap is an example of

- a) complete binary tree
- b) spanning tree
- c) sparse tree
- d) binary search tree

4. What is the time complexity for inserting/deleting at the beginning of the array?

- a)  $O(1)$
- b)  $O(n)$
- c)  $O(\log n)$
- d)  $O(n \log n)$

5. Visiting root node after visiting left and right sub-trees is called

- a) In-order Traversal
- b) Pre-order Traversal
- c) Post-order Traversal

6. What is the time complexity to count the number of elements in the linked list?

- a)  $O(1)$
- b)  $O(n)$
- c)  $O(\log n)$
- d) None of the mentioned

P.T.O.

7. What does 'stack underflow' refer to?
- a) accessing item from an undefined stack
  - b) adding items to a full stack
  - c) removing items from an empty stack
  - d) index out of bounds exception
8. Heap can be used as \_\_\_\_\_
- a) Priority queue
  - b) Stack
  - c) A decreasing order array
  - d) None of the mentioned
9. Which of the following concepts make extensive use of arrays?
- a) Binary trees
  - b) Scheduling of processes
  - c) Caching
  - d) Spatial locality
10. The Big O of bubble sort is:
- a)  $O(n)$
  - b)  $O(\log n)$
  - c)  $O(n^2)$
  - d) None of these



**UNIVERSITY OF THE PUNJAB**  
**B.S. 4 Years Program / Fourth Semester – 2019**

Roll No. ....

Paper: Data Structure and Algorithm  
 Course Code: IT-207 / IT-22408 Part – II

Time: 2 Hrs. 45 Min. Marks: 50

**ATTEMPT THIS (SUBJECTIVE) ON THE SEPARATE ANSWER SHEET PROVIDED**

**Question # 2. Short Questions: (4 x 5 =20 marks)**

- a) What are the time complexities of each of the following code segments? Write your answers in Big Oh notation only.

```
for(int i = 1; i <= n; i++)
{
    for(int j = 0; j <= 10000; j++)
    {
        temp++;
    }
}
```

- b) Apply the Quick sort algorithm on this array and show the contents after one iteration (partitioning this array once). Use the first number (12) as the pivot.

12	2	15	11	25	3	5	18	8	6
----	---	----	----	----	---	---	----	---	---

- c) Write down the post-fix forms of the following infix expressions (keeping in mind the operator precedence rules of C++):

$$A + C + B / D - E * F / G$$

- d) Write brief answers to the following two questions:
- o Name one advantage of a linked list over an array.
  - o Name one advantage of an array over a linked list.

**Question # 3. Long Questions (2 x 15 = 30 marks)**

- a) Given the following declaration of a Circular-Doubly-Linked-List (with a dummy header node) which stores integers in unsorted order:

<pre>class DNode {     friend class CDLinkedList; private:     int data;     DNode* next;     DNode* prev; };</pre>	<pre>class CDLinkedList { private:     DNode head; public:     void insertAtEnd (int val); };</pre>
---	---

You are required to implement the following member function `void insertAtEnd(int val)` of the `CDLinkedList` class. This function inserts "val" at the end of the linked list. The time complexity of this function should be constant i.e.  $O(1)$ .

- b) Write a recursive C++ function to count all occurrences of a given number `k` in a given array of integers. The prototype of your function should be:

```
int count (int* arr, int start, int end, int k)
```

In the above prototype, `arr` is the array, `start` and `end` are starting and ending indices of the array, and `k` is the number whose occurrences are to be counted.

For example, if the array `arr` contains these 10 integers {2,1,7,5,7,2,6,7,8,7} and we call `count(arr, 0, 9, 7)` then this function should return 4 (because the number 7 occurs 4 times in the above array).

*Note: Do NOT change the function prototype. Do NOT use any global/static variables. The implementation must be recursive. No Marks for iterative implementation.*