Name: _____Reg No. _____ Scores_____

CP 213: DATA STRUCTURES AND ALGORITHMS ANALYSIS

TEST 1 **15 Marks** **DATE _____**

Instruction: Answer ALL questions provided

Question One

Choose the most correct answer from the given options. **(0.5 Marks Each)**

i. Which of the following best describes why insertion at the tail of a singly linked list can be O(1) in some implementations?

A. Because singly linked lists always store elements in contiguous memory.

B. Because the list keeps a pointer/reference to the tail node.

C. Because nodes store explicit indices for fast access.

D. Because the list uses an underlying array and resizes automatically.

ii. In C, what does the expression a[i] mean when array a is declared as int a[10]?

A. The value stored at memory address (a + i*sizeof(int))*.

B. The pointer to the i-th element (type int).

C. The i-th element value (an int), equivalent to *(a + i).

D. The size of the array minus i.

iii. Which of the following is a correct statement about recursion vs iteration?

A. Any recursive algorithm cannot be expressed iteratively.

B. Recursive algorithms always use less memory than iterative ones.

C. Recursion naturally models problems with recursive structure (e.g., trees).

D. Iterative solutions always have better asymptotic time complexity than recursive ones.

iv. What is the main advantage of using a linked list over an array?

A. Faster random access by index.

B. Constant-time insertion/deletion at known positions (with links), and dynamic size without reallocation.

C. Always uses less memory than arrays.

D. Sorting linked lists is always faster than arrays.

v. Consider a singly linked list with only head pointer (no tail pointer). What is the time complexity to add an element to the end?

A. O(1)

B. O(log n)

C. O(n)

D. O(n^2)

vi. Which snippet best expresses the base case for the recursive computation of factorial(n)?

A. if n < 2 return 1

B. if n == 0 return 0

C. if n > 1 return n * factorial(n-1)

D. if n == 1 return n + 1.

vii. In C/C++, which of these is a valid reason to use pointers?

A. They enforce bounds checking automatically.

B. They allow dynamic memory and references to other objects.

C. They convert integers into floats.

D. They prevent segmentation faults.

viii. What is the result of applying pop to an empty stack (array implementation) if not checked?

A. A well-defined zero returned.

B. Underflow / undefined behavior.

C. Overflow error.

D. Memory compaction.

ix. Which array operation is worst if you must maintain order and use a fixed-size array?

A. Read by index.

B. Append at end (if space)

C. Insert at front (shift all elements)

D. Check length if stored

x. Which queue operation should you use to just peek at the front without removing it?

A. dequeue().

B. enqueue().

C. peek() / front().

D. pop().

Question Two

Match items in column B with its most proper explanation from column A.

(0.5 Marks Each)

Column A	Column B
i. Linear data structure.	A. Queue
ii. Accessing the second element of an array "a" based on the pointer operator.	B. Push
	C. Caching
iii. It is the condition that occurs when a pop( ) operation is called on an empty queue.	D. n – 1
	E. *a + 2
iv. A sorting algorithm that provides the best time complexity in the worst-case scenario.	F. Array.
	G. Insert,
v. The maximum number of swaps that can be performed in the Selection Sort algorithm.	H. Underflow.
	I. Bubble Sort.

Name: _____ Reg No. _____ Scores_____

vi. The time complexity of the binary search algorithm. vii. Highly uses the concept of an array. viii. Inserting an element in the stack. ix. The principle under which Queue operates. x. It is the linear data structure in which insertion and deletion operations can be performed from both ends.	J. *(a + 2). K. FIFO. L. Garbage Value. M. Merge Sort. N. n – 2. O. O(log n). P. Deque. Q. Add. R. Spatial Locality. S. LIFO T. *(a +1) U. *a + 1

Question Three

Write T for the correct and F for the incorrect statements. **(0.5 Mark Each)**

i. The new keyword frees the memory allocated dynamically.
ii. Push adds items item into a stack.
iii. Infix expression has an operator between operands.
iv. Void is the type with an empty set of values.
v. Enqueue is a stack operation.
vi. LIFO is a working principle of a tree data structure.
vii. Return keyword terminates the current function and returns the specified value (if any) to the caller.
viii. The size of an array is calculated by the sizeof operator.
ix. /n is a newline operator.
x. The delete keyword assigns a new value to memory.