**Student Id: Solution Section: BAI-3A**\_\_\_\_\_\_  **Marks: 15**

**Quiz#01 - Paper B**

**Question#01**

Consider the content of an integer array as below:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] |
| 2 | 43 | 0 | 3 | 1 | 11 | 12 | 1 | 10 | 21 |

How the content of the array will look like after the 5th iteration of the outer loop in the following cases:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Insertion Sort | | | | | | | | | |
| [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] |
| 0 | 1 | 2 | 3 | 11 | 43 | 12 | 1 | 10 | 21 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Selection Sort | | | | | | | | | |
| [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] |
| 0 | 1 | 1 | 2 | 3 | 11 | 12 | 43 | 10 | 21 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bubble Sort | | | | | | | | | |
| [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] |
| 0 | 1 | 1 | 2 | 3 | 10 | 11 | 12 | 21 | 43 |

**Question#02**

You are given a singly linked list with the following values:

15 -> 9 -> 12 -> 6 -> 11 -> 3 -> 14 -> 1

Answer the following questions:

1. Identify the sorting algorithm being used to sort this singly linked list.

Bubble sort

1. Write the time complexity of the sorting algorithm.

O(n2)

Dry Run:

|  |
| --- |
| **Initial List:**  15 -> 9 -> 12 -> 6 -> 11 -> 3 -> 14 -> 1  **List after first iteration:**  9 -> 15 -> 12 -> 6 -> 11 -> 3 -> 14 -> 1  **List after second iteration:**  9 -> 12 -> 15 -> 6 -> 11 -> 3 -> 14 -> 1 |

**Question#03**

1. You are given the head of a linked list and two integers, k1 and k2. Return the head of the linked list after swapping the values of the nodes at the k1-th and k2-th positions from the beginning (the list is 1-indexed).

**Input:**

* Head: 1 -> 2 -> 3 -> 4 -> 5
* k1: 2
* k2: 4

**Output:**

* Head: 1 -> 4 -> 3 -> 2 -> 5

|  |
| --- |
| Node\* swapNodes(Node\* head, int k1, int k2) {  if (k1 == k2) return head;  Node\* k1\_node1 = nullptr;  Node\* k2\_node2 = nullptr;  Node\* current = head;  int count = 1;  while (current != nullptr) {  if (count == k1) k1\_node1 = current;  if (count == k2) k2\_node2 = current;  current = current->next;  count++;  }  if (k1\_node1 != nullptr && k2\_node2 != nullptr) {  int temp = k1\_node1->val;  k1\_node1->val = k2\_node2->val;  k2\_node2->val = temp;  }  return head;  } |

1. You are given the head of a singly linked list. Write a function to delete all nodes whose values are less than a specified integer x. After deleting these nodes, the function should return the head of the updated linked list.

### Input:

* Head: 5 -> 3 -> 8 -> 1 -> 7 -> 2 -> 6
* x: 4

### Output:

* Head: 5 -> 8 -> 7 -> 6

|  |
| --- |
| Node\* removeNodesLessThan(Node\* head, int x) {  Node dummyNode(0);  dummyNode.next = head;  Node\* currentNode = &dummyNode;  while (currentNode->next != nullptr) {  if (currentNode->next->value < x) {  Node\* nodeToDelete = currentNode->next;  currentNode->next = currentNode->next->next;  delete nodeToDelete;  } else {  currentNode = currentNode->next;  }  }  return dummyNode.next;  } |