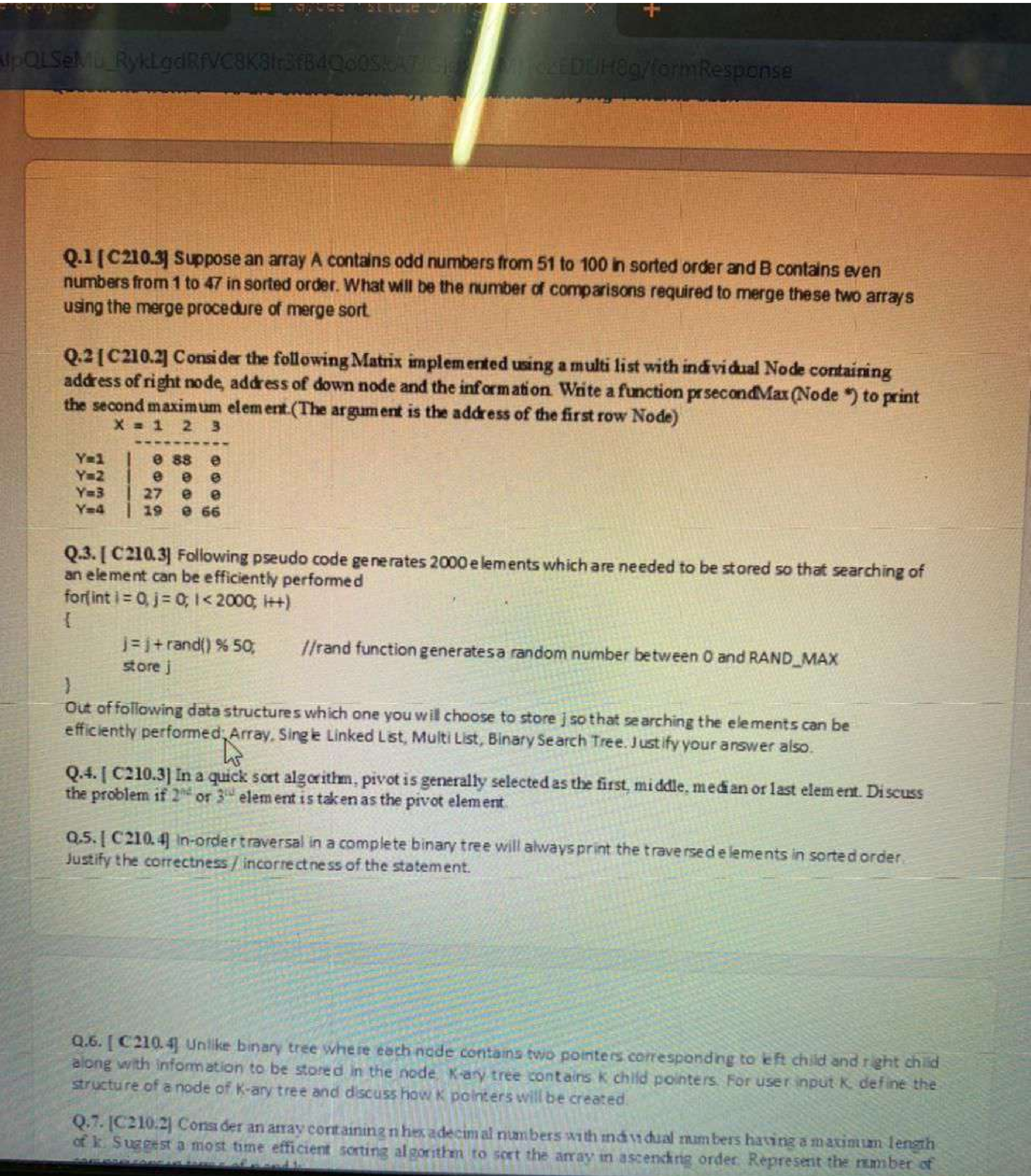


**Q.1 [ C210.3]** Suppose an array A contains odd numbers from 51 to 100 in sorted order and B contains even numbers from 1 to 47 in sorted order. What will be the number of comparisons required to merge these two arrays using the merge procedure of merge sort

**Q.2 [ C210.2]** Consider the following Matrix implemented using a multi list with individual Node containing address of right node, address of down node and the information. Write a function prsecondMax (Node *) to print the second maximum element.(The argument is the address of the first row Node)

```
       X = 1  2  3
          ----------
Y=1   |   0 88  0
Y=2   |   0  0  0
Y=3   |  27  0  0
Y=4   |  19  0 66
```

**Q.3. [ C210.3]** Following pseudo code generates 2000 elements which are needed to be stored so that searching of an element can be efficiently performed

```
for(int i = 0, j = 0; i < 2000; i++)
{
        j = j + rand() % 50;        //rand function generates a random number between 0 and RAND_MAX
        store j
}
```

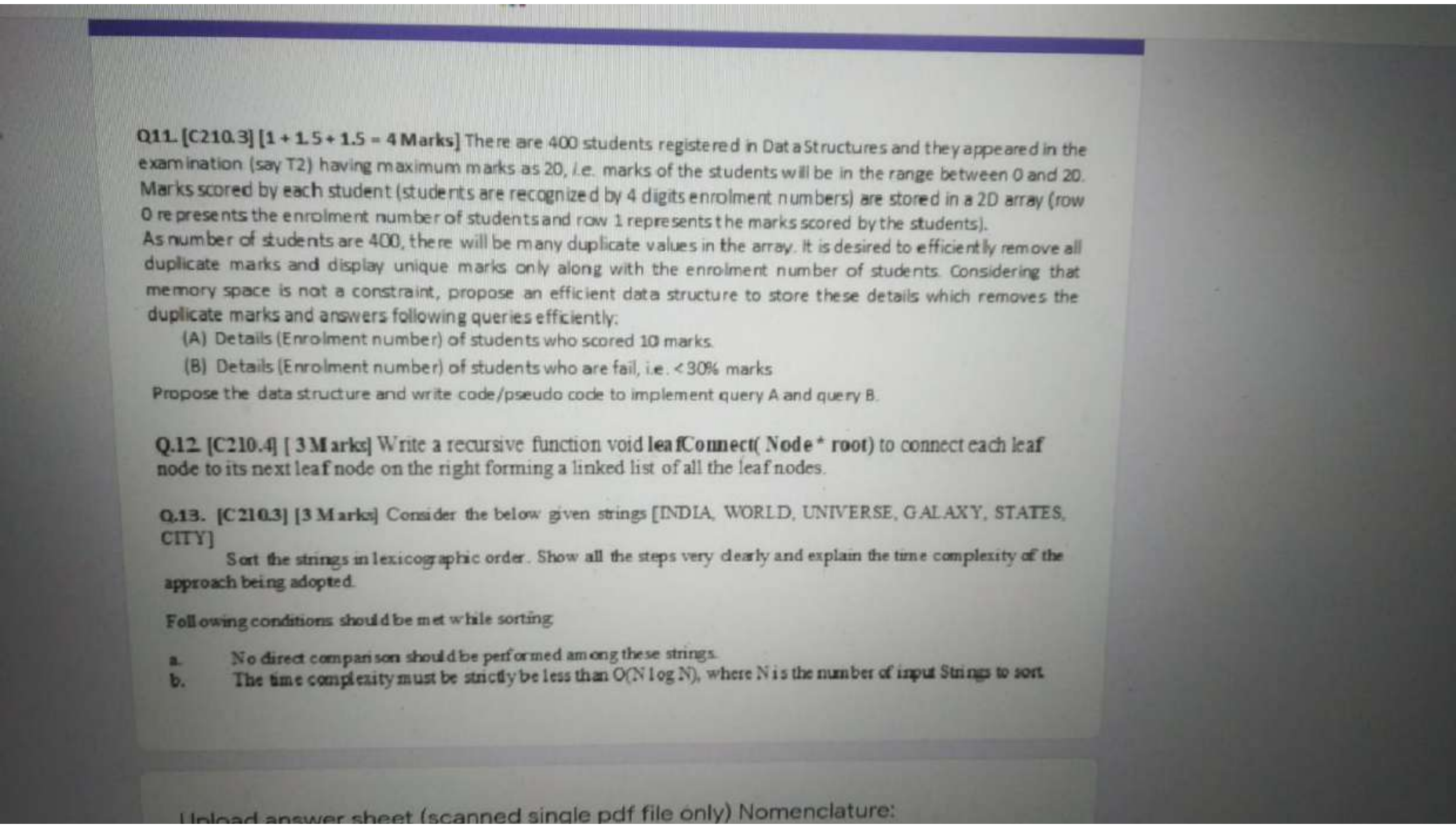
Out of following data structures which one you will choose to store j so that searching the elements can be efficiently performed: Array, Single Linked List, Multi List, Binary Search Tree. Justify your answer also.

**Q.4. [ C210.3]** In a quick sort algorithm, pivot is generally selected as the first, middle, median or last element. Discuss the problem if 2nd or 3rd element is taken as the pivot element

**Q.5. [ C210.4]** In-order traversal in a complete binary tree will always print the traversed elements in sorted order. Justify the correctness / incorrectness of the statement.

**Q.6. [ C210.4]** Unlike binary tree where each node contains two pointers corresponding to left child and right child along with information to be stored in the node. K-ary tree contains K child pointers. For user input K, define the structure of a node of K-ary tree and discuss how K pointers will be created

**Q.7. [C210.2]** Consider an array containing n hexadecimal numbers with individual numbers having a maximum length of k. Suggest a most time efficient sorting algorithm to sort the array in ascending order. Represent the number of



**Q11. [C210.3] [1 + 1.5 + 1.5 = 4 Marks]** There are 400 students registered in Data Structures and they appeared in the examination (say T2) having maximum marks as 20, *i.e.* marks of the students will be in the range between 0 and 20. Marks scored by each student (students are recognized by 4 digits enrolment numbers) are stored in a 2D array (row 0 represents the enrolment number of students and row 1 represents the marks scored by the students).
As number of students are 400, there will be many duplicate values in the array. It is desired to efficiently remove all duplicate marks and display unique marks only along with the enrolment number of students. Considering that memory space is not a constraint, propose an efficient data structure to store these details which removes the duplicate marks and answers following queries efficiently:

(A) Details (Enrolment number) of students who scored 10 marks.

(B) Details (Enrolment number) of students who are fail, i.e. < 30% marks

Propose the data structure and write code/pseudo code to implement query A and query B.

**Q.12 [C210.4] [ 3 Marks]** Write a recursive function void **leafConnect( Node * root)** to connect each leaf node to its next leaf node on the right forming a linked list of all the leaf nodes.

**Q.13. [C210.3] [3 Marks]** Consider the below given strings [INDIA, WORLD, UNIVERSE, GALAXY, STATES, CITY]

Sort the strings in lexicographic order. Show all the steps very clearly and explain the time complexity of the approach being adopted.

**Following conditions should be met while sorting**

a. No direct comparison should be performed among these strings.

b. The time complexity must be strictly be less than O(N log N), where N is the number of input Strings to sort

Upload answer sheet (scanned single pdf file only) Nomenclature: