POSSESION OF MOBILES IN EXAMS IS UFM PRACTICE.

NAME Vishwakant Enrollment No. 2110[redacted]

Jaypee Institute of Information Technology, Noida

T1 Examination, Even 2023

B. Tech. IV Semester

Course Name: Algorithms and Problem Solving
Course Code: 15B11CI411

Maximum Time: 1 Hr.
Maximum Marks: 20 Marks

CO1	Analyse the complexity of different algorithms using asymptotic analysis.
CO2	Select appropriate sorting and searching technique for problem solving
CO3	Apply various algorithm design principles for solving a given problem
CO4	Identify, formulate and design an efficient solution to a given problem using appropriate data structure and algorithm design technique.

Q1. [CO2] [5 Marks] You have been given a two-dimensional array, A[0 .. m-1, 0 .. n-1] in which every row and every column are sorted. and an integer k. Propose an efficient algorithm to find the k^{th} biggest element in the given two-dimensional array, A.

Q2. [CO1] [2 Marks] In context of Quick Sort, formulate the recurrence for following scenarios:

(a) Pivot always lands in the beginning of the array
(b) Pivot always lands in the middle of the array
(c) Pivot always lands in the middle third (*i.e.* pivot is at one third position) of the array

Q3. [CO3] [6 Marks] Your friend, ABC has a collection of N lock-boxes (labelled as B_1, B_2, ... B_N) and labelled keys (K_1, K_2...K_N) to unlock or lock the respective boxes (key, K_i to lock/unlock the box B_i). However, quantity of some keys is more than one (*i.e.* duplicate keys are also there), *viz.*, as observed from Table 1, quantity of the key, K_1 (which will lock/unlock the box, B_1) is 3. In all, there are M keys, where, M > N. For personal reasons, ABC keeps the keys inside these N boxes and keeps the record, R of the keys kept in different boxes, *i.e.* which key is kept in which box (an example of record, R is given in Table 1). One box may contain one or more than one keys, however, the key, K_i which is used to lock/unlock the box, B_i cannot be kept in B_i, *viz.* key, K_1 cannot be kept in box, B_1. Your friend gave you the record R, all the N boxes (which are initially locked) and a duplicate key K_1, which will be used to unlock the box, B_1. Starting with B_1, your task is to unlock all the boxes and return back the keys of all boxes to ABC. Out of several keys kept in B_1, you are allowed to pick only one key, say K_i which will be used to open the box B_i. Further, from box, B_i you are allowed to pick only one key, say K_j which will be used to open the box B_j, and so on, *i.e.* you keep on collecting **exactly one key from each box** after opening them. When you have collected all the keys (total N+1 keys, remember that the key for box, B_1 was initially given to you, so you will have two keys (K_1) at the end), you need to handover N+1 keys to ABC.

Propose an efficient scheme to find out the sequence of boxes to be opened so that you can handover N+1 keys to your friend. Further, you have to apply the proposed scheme to the record, R (detailing the boxes and the keys stored in these boxes) given in Table 1 to find out the sequence (starting with B_1) of boxes to be opened so that you can handover all the keys (one key each for B_2 to B_{10} and 2 keys for B_1) to your friend.

Table 1: Record, R – List of Boxes and the Keys stored in these boxes

Box Number	Keys kept in the Box (key, K_i to be used to open box B_i)
B_1	K_2, K_8, K_9
B_2	K_1, K_3, K_9
B_3	K_2, K_4, K_9, K_{10}
B_4	K_3, K_5, K_{10}
B_5	K_1, K_6, K_{10}
B_6	K_5, K_7, K_{10}
B_7	K_6, K_8, K_9, K_{10}
B_8	K_1, K_7, K_9
B_9	K_1, K_2, K_3, K_7, K_8, K_{10}
B_{10}	K_3, K_4, K_5, K_6, K_7, K_9

Q4. [CO1] [3 Marks] While computing the Big Oh (O) for following functions, what will be the g(n) and constants, C and no for respective functions:

(i) $f(n) = 3n^4 + 2n + 6$

(ii) $f(n) = 2n^3 + 4n^2 + n + 8$

Q5. [CO1] [4 Marks] A function, f1() is given in Fig. 1. Answer following related to f1().

(a) Obtain the recurrence for the function, f1(), and solve it using appropriate recurrence solving scheme.

(b) Considering the array, arr as sorted array, mention the purpose/utility of the function f1(). Further, for arr (sorted array) as {2, 5, 8, 9, 12, 15, 19, 23, 32, 36}, v1 as 0, v2 as 9, and v3 as 4, compute the output of the function f1().

```
int f1(int *arr, int v1, int v2, int v3)
{
    int v4   = v1 + (v2 - v1) / 3;
    int v5   = v2 - (v2 - v1) / 3;
    if (arr[v4] == v3)
        return v4;
    else if (arr[v5] == v3)
        return v5;
    if(v1 < v2)
    {
        if (arr[v4] > v3)
            f1(arr, v1, v4 - 1, v3);
        else if (arr[v5] < v3)
            f1(arr, v5 + 1, v2, v3);
        else
            f1(arr, v4 + 1, v5 - 1, v3);
    }
    else
        return -1;
}
```

Fig 1: The function, f1()