End-Semester Lab. Examination, July-2022
Algorithm Design-2 (CSE 4131)

Semester: 4th
Full mark: 15

Branch: CSE, CS&IT
Time: 90 Mins.

All questions are compulsory.

Q1. Give the Java/C/C++/Python code implementation of the following problem.
Weighted interval scheduling Problem using dynamic programming.

Q2. Using backtracking, let us generate all possible subsets of a given set $S = \{71, 51, 91\}$, using the code given in section-7.1.1 of book (i.e. The Algorithm Design Manual by Steven S. Skiena). In how many number of steps the subset $\{71, 91\}$ will be generated and in that step what are the contents of k and $c[i]$? Demonstrate the steps neatly.

```
generate_subsets(int n){
    backtrack(a[],0,n);
}

backtrack(int a[],int k,int n) {
    if(is_a_solution(a[],k,n))
        process_solution(a[],k,n);
    else {
        k = k+1;
        construct_candidates(a[],k,n,c,&nc);
        for(i=0; i<nc; i++) {
            a[k] = c[i];
            make_move(a[],k,n);
            backtrack(a[],k,n);
            unmake_move(a[],k,n);
            if(finished) return; // finished = FALSE
        }
    }
}
```

End-Semester Lab. Examination, July-2022
Algorithm Design-2 (CSE 4131)

Q3. Let A be an N × N two-dimensional (2D) array with all distinct elements, in which all rows and all columns are sorted in ascending order from smaller to larger indices. Given a key K, your task is to find out whether K is present in this 2D array A.

Propose a recursive algorithm to solve this, from which you can design a $\Theta(n \log_2 n)$-time algorithm.

End of Questions

Instructions:

The evaluation will be done in the following ways:
- Correct implementation with satisfactory response to on-spot questions: 5 / 5
- Correct implementation with unsatisfactory response to on-spot questions: 3 / 5
- Incorrect/partial (min. 80%) implementation with satisfactory response to on-spot questions: 3 / 5
- Incorrect/partial (min. 80%) implementation with unsatisfactory response to on-spot questions: 2 / 5
- No implementation with satisfactory response: 1.5 / 5
- No implementation with unsatisfactory response: 0.5 / 5
- Plagiarized code: -2.5 / 5