# CSE 4131: ALGORITHM DESIGN – II

## ASSIGNMENT 2:

This assignment is designed to give you practice with:

- Dynamic Programming (SECTION – A)
- Combinatorial Searches and Heuristic methods (SECTION – B)

You are allowed to use only those concepts which are covered in the lecture class till date.

Plagiarized assignments will be given zero mark.

## Submission deadline: 25th October, 2018 (6PM)

---

SECTION – A

---

We have covered/discussed the following topics under Dynamic Programming:

- Generating n$^{th}$ Fibonacci number
- Computing binomial coefficients
- Approximate string matching (using edit distance)
- Longest Increasing Subsequence
- Longest Common Subsequence
- The (Linear) Integer Partition problem
- Parsing context-free-grammars (CYK algorithm)
- Minimum weight triangulation
- Matrix chain multiplication

***You have to implement any five of the above dynamic programming algorithms using Java/C/C++ code. Submit the soft copy of the code to your respective class teacher's official mail id, if and only if you can explain the code in viva-voice. No need to submit the code, if you have no idea about the program. If you will submit and fail in explaining in the viva then you will awarded with negative marks.***

<u>Solve the following questions and submit in handwritten hard copy.</u>

1. Consider two strings A = “qpqrr” and B = “pqprqrp”. Let x be the length of the longest common subsequence (not necessarily contiguous) between A and B and let y be the number of such longest common subsequences between A and B. Determine x and y using dynamic programming. Then determine x + 10y .

2. Given a CNF grammar G:

S → AB | BC

A → BA | a

B → CC | b

C → AB | a

and string w is "baaba".

Show using CYK Algorithm whether w is in L(G) or not. Construct the table using dynamic programming.

3. Determine a *longest common subsequence (LCS)* of ⟨1,0,0,1,0,1,0,1⟩ and ⟨0,1,0,1,1,0,1,1,0⟩.

4. Prove that every triangulation of an $n$-vertex convex polygon has $n - 3$ chords and divides the polygon into $n - 2$ triangles.

5. Give an optimal parenthesization of a matrix-chain product whose sequence of dimension is ⟨4,2,7,1,5,3⟩.

---

## SECTION - B

---

We have covered/discussed the following topics under Combinatorial search and backtracking:

- Constructing all subsets
- Constructing all permutations
- Constructing all paths in a graph
- Sudoku problem
- N-queen problem
- Sum-of-Subset
- Search pruning
- Heuristic search methods

1. Let us generate all possible subsets of a given set S = {7,5,9}, using the code given in section-7.1.1 of our text book (i.e. The Algorithm Design Manual by Steven S. Skiena). In how many number of steps the subset {a,d} will be generated and in that step what are the contents of k and c[i]? Answer this with complete demonstration of the steps being executed.

2. Let us generate all possible permutations of a given set S = {7,5,9}, using the code given in section-7.1.2 of our text book (i.e. The Algorithm Design Manual by Steven S. Skiena). In how many number of steps the arrangement {a,d,b} will be generated and in that step what are the contents of k, in_perm[i] and in_perm[a[i]]? Answer this with complete demonstration of the steps being executed. Draw the search-space-tree displaying all possible arrangements.

3. There are 5 distinct numbers {1,2,5,6,8}. Find the combinations of these numbers such that the sum is 9. Use *backtracking* model to arrive at the solution.

4. The 8-puzzle problem: It is played on a 3-by-3 grid with 8 square blocks labeled 1 through 8 and a blank square. Your goal is to rearrange the blocks so that they are in order. You are permitted to slide blocks horizontally or vertically into the blank square(i.e. x). The following shows a sequence of legal moves from an initial board position (left) to the goal position (right).

```
 x 1  3         1  x  3         1  2  3         1  2  3         1  2  3

4  2  5   =>   4  2  5   =>   4  x  5   =>   4  5  x   =>   4  5  6

7  8  6        7  8  6        7  8  6        7  8  6        7  8  x

 initial                                                      goal
```

(a) Identify the implicit and explicit constraints for a backtracking formulation.

(b) Draw the state-space-diagram for the above problem.

(c) How many states are there? How many (legal) transitions are there?

5. Consider the following 4X4 Sudoku problem.

| | | | |
|---|---|---|---|
| 1 | 3 | | |
| | | 2 | 1 |
| | 1 | | 2 |
| 2 | 4 | | |

(a) In solving the Sudoku problem using backtracking, what are the common heuristics used to identify the first empty cell?

(b) Solve the above Sudoku using both the heuristics.

(c) Compare the number of backtrack moves used in both the techniques.

6. **N-Queen Problem:** Given a chess board having N×N cells, we need to place N queens in such a way that no queen is attacked by any other queen. A queen can attack horizontally, vertically and diagonally.

So initially we are having N×N unattacked cells where we need to place N queens. Let's place the first queen at a cell (i,j), so now the number of unattacked cells is reduced, and number of queens to be placed is N−1. Place the next queen at some unattacked cell. This again reduces the number of unattacked cells and number of queens to be placed becomes N−2. Continue doing this, as long as following conditions hold.

- The number of unattacked cells is not 0.
- The number of queens to be placed is not 0.

If the number of queens to be placed becomes 0, then it's over, we found a solution. But if the number of unattacked cells become 0, then we need to backtrack, i.e. remove the last placed queen from its current cell, and place it at some other cell. We do this recursively.

(a) Give a recursive backtracking algorithm for this N−queen problem.

(b) Demonstrate the execution steps for n=4 in the algorithm given in 7(a).

(c) Draw the state−space−diagram to display all possible solutions for n=4.

(d) List the implicit and explicit constraints in this problem.

7. Give a comparative study between random sampling, local search and simulated annealing.