UE19CS251
Design and Analysis of Algorithms
Question Bank (Unit 5)

PES University

- **Dynamic Programming**
  - What does dynamic programming have in common with divide-and-conquer? What is a principal difference between them?
  - The coin change problem does not have an optimal greedy solution in all cases (*ex:* coins 1,20,25 and amount 40). Is there a dynamic programming based algorithm that can solve all cases of the coin change problem?
- **The Knapsack Problem**
  - Write pseudocode of the bottom-up dynamic programming algorithm for the knapsack problem
  - True or False:
    1. A sequence of values in a row of the dynamic programming table for the knapsack problem is always nondecreasing?
    2. A sequence of values in a column of the dynamic programming table for the knapsack problem is always nondecreasing?
- **Memory Function Knapsack**
  - Consider the use of the MF technique to compute binomial coefficient using the recurrence

$$C(n, k) = C(n-1, k-1) + C(n-1, k)$$

    * How many table entires are filled?
    * How many are reused?
- **Transitive Closure (Warshall's Algorithm)**

- Is Warshall's algorithm efficient for sparse graphs? Why / why not?
- Can Warshall's algorithm be used to determine if a graph is a DAG (Directed Acyclic Graph)? If so, how?

- **All Pairs Shortest Path (Floyd's Algorithm)**
  - Give an example of a graph with negative weights for which Floyd's algorithm does not yield the correct result
  - Enhance Floyd's algorithm so that shortest paths themselves, not just their lengths, can be found

- **Optimal Binary Search Trees**
  - Space: $\Theta(n^2)$
  - Time: $\Theta(n^3)$, can be reduced to $\Theta(n^2)$ by exploiting monotonicity of entries in root table ($R[i, j]$ is always between $R[i, j-1]$ and $R[i+1, j]$
  - Method can be expanded to include unsuccessful searches

- **Lower-Bound Arguments**
  - Prove that the classic recursive algorithm for the Tower of Hanoi puzzle makes the minimum number of disk moves
  - Find a trivial lower-bound class and indicate if the bound is tight:
    - finding the largest element in an array
    - generating all the subsets of an n-element set
    - determining whether n given real numbers are all distinct

- **Decision Trees**
  - Consider the problem of finding the median of a three-element set a, b, c of orderable items
    - What is the information-theoretic lower bound for comparison-based algorithms solving this problem?
    - Draw a decision tree for an algorithm solving this problem
    - Is the above bound tight?