Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam - 603 110
(An Autonomous Institution, Affiliated to Anna University, Chennai)

UCS2403: DESIGN & ANALYSIS OF ALGORITHMS

Assignment 9

1. Using adjacency matrix representation for the input graph, implement the following algorithms:
   - Dijkstra's algorithm
   - Prim's algorithm
   - Kruskal's algorithm

2. The job scheduling problem takes as input a set of jobs with deadlines and profits. A subset of jobs with maximum profit is the final output. Develop and implement a greedy algorithm to solve the job scheduling problem.

3. Consider the knapsack problem studied in the class.

   (a) Write a Python code to implement the following greedy strategy. Pick the item that has the maximum price per unit weight and add to the knapsack. Continue adding items until no more items can be added to the knapsack (adding any more results in exceeding the knapsack capacity). Give a counterexample (different from the one taught in the class) to show that this strategy may not return an optimal solution all the time.

   (b) Write the Python code to implement a dynamic programming algorithm for the knapsack problem.

4. Consider the problem of making change for n rupees using the smallest number of coins. Assume that each coin's value is an integer.

   (a) Describe a greedy algorithm to make change consisting of ₹1, ₹5, ₹10, and ₹25 (hypothetical) coins. Does your algorithm yield an optimal solution? Implement the algorithm in Python.
   Example: If you need to make change for ₹126, then you need to pick five ₹25 coins and one ₹1 coin.

   (b) Modify the greedy algorithm such that the coin denominations are powers of c, where $c > 1$. That is, the denominations of coins available are $c^0, c^1, \ldots, c^k$. Does this algorithm give an optimal solution? Implement the algorithm in Python.

(c) Give a set of coin denominations for which the greedy algorithm does not yield an optimal solution. Your set should include ₹1 so that there is a solution for every value of n (not necessarily optimal).

(d) Give an algorithm (not necessarily greedy) that makes change for any set of k different coin denominations using the smallest number of coins, assuming that one of the coins is a ₹1 coin. Implement the code in Python. What is the time complexity of the algorithm as a function of n alone or as a combination of n and k?