19CSE302 – Design and Analysis of Algorithms

Lab Assignment 4

Solve the problems using Greedy Approach.

1. Solve the fractional knapsack problem for the given set of N items with benefit-weight pairs and sack capacity of W. Print the optimal solution. Use the examples discussed in the class.
2. Given an array of N integer, we have to maximize the sum of arr[i] * i, where i is the index of the element (i = 0, 1, 2, ..., N). We can rearrange the position of the integer in the array to maximize the sum.

   - Time complexity of Brute Force Approach: O(N x N!)
   - Time complexity of Greedy Algorithm: O(N logN)

   For example, consider the following array:
   - arr[] = {2, 5, 3, 4, 0}
   - In this arrangement, the sum of products will be:

   - 2 * 0 + 5 * 1 + 3 * 2 + 4 * 3 + 0 * 5
   - = 0 + 5 + 6 + 12 + 0
   - = 23

   To maximize the sum we have to arrange the array as [0, 2, 3, 4, 5]

   - So, the sum will be

   - 0 * (0) + 2 * (1) + 3 * (2) + 4 * (3) + 5 * (4)
   - = 0 + 2 + 6 + 12 + 20
   - = 40

   So, 40 is the maximum for the given array.

3. Given two arrays array_One[] and array_Two[] of same size N. We need to first rearrange the arrays such that the sum of the product of pairs (1 element from each) is minimum. That is SUM (A[i] * B[i]) for all i is minimum.
   - Time complexity of Brute Force Approach: O((N!)^2)
   - Time complexity of Greedy Algorithm: O(N logN)

For example, consider the following two arrays:

array_one[] = {7, 5, 1, 4};

array_two[] = {6, 17, 9, 3};

If we arrange the array_one like {1, 4, 5, 7} and array_two like {17, 9, 6, 3}

Then the sum of products is: (17 * 1) + (9 * 4) + (6 * 5) + (7 * 3) = 17 + 36 + 30 + 21 = 104 which is the minimum sum of products.

The minimum sum of product is 104.