

**CSE2046- Analysis of Algorithms Homework #2**

Authors:

*Ahmet Hakan Şimşek - 1501170601*

*İsmail Ertan Sarıhan - 150116036*

*Oğuzhan Karabudak - 150118005*

# INTRODUCTION

In this project we are assigned with multi constrained knapsack problem which is extension of knapsack problem. In this problem we have a volume limit and weight limit where volume and weight of each item is not related. If we have m items and n knapsacks we have m.n different weight amount which is a w(i,j). Also if we insert an item to knapsack, we have to put copy of that item to each knapsack simultaneously. Our aim is to maximize total value of selected item by obeying the capacity constraints.

**Multi-constraint Knapsack Problem**

Multi-constraint Knapsack problem is NP-Hard problem which means it is diffucult to find optimal solutions especially for large instances because up to now no one could not find any polinomial time algorithm to solve knapsack problem. So we search so much in internet to find heuristic algorithms to solve this problem. At the end we tried to use first greedy algorith to start with a string which consist of 0s and 1s after we tried to improve our this string to generate better sequences randomly.We described details of algorithm below.

**Our Approach**

Our algorithm consist of mainly 4 parts; input part, greedy part, generated random string part and printing output part.

1-) Input part: We take according to pdf document shared classroom and this part’s algorithm find properly it imports from txt file and reads at most 10 values per line and load them into proper arrays to use future purposes( here one of friends misunderstand and created object for OOP but then we put them into array this look maybe different for you but we did not used object)

2-) Greedy part: We calculated value per weight for each item bu dividing values to weight for each item and knapsack and sum to totalValuePerWeight. This array is sum of an item’s value per weight for all knapsacks. We are going to put which item has the highest value in this array. By appling this greedy algorithm we have a string of 0s and 1s.

3-) Generated random string part: Here we tries to find better value for our knapsacks by generating random strings but we have used previous greedy string. First we pick some items randomly and then after this process we tried to put item as much as we could and at the end we compared new knapsack value and previous knapsack value, if new one is greater we are going to use new algorithm.

4-) Printing output part: Here we are at the end of the algorith, we have found highest value in previous part, now it is time to print it to output txt file. We did this part according to pdf document shared classroom.

As a result, we have tried this algorithm over other knapsack inputs that we have found in internet and we believe that our algorithm work really good because result of other multi constraint knapsack problem are lower than %0.1 error margin also this algorithm work in a good amount of time.

150117060 Ahmet hakan Şimşek

150116036 Ertan sarıhan

150118005 Oguzhan karabudak