Muhammad Umar 05721

Lab 08

# Introduction:

In this lab I’ve implemented the famous coin change algorithm using greedy and dynamic programming approaches. I’ve implemented it in python.

# Approach and Description:

Greedy algorithm always starts with the coin having largest value. It divides the amount with the coin value hence number of coins are obtained. The product of value of the coin and the number of coins, is subtracted from the amount. The same process is repeated with the remaining amount. Hence number of coins for the change are obtained.

For dynamic algorithm, it divides the amount into different overlapping problems and then solves them and stores their solution in a list, whenever a problem occurs whose solution has been computed earlier, its solution is not computed again, rather it is retrieved from the list in which we’ve stored it earlier.

File named “dynamic.py” and “greedy.py” contains functions for dynamic approach and greedy approach respectively. Both functions take two arguments. First argument is money or the amount of cash whose change is needed. The second argument is the list of denominations or the list of coins we have to make this change with. Both functions return the number of coins required to make the change of the given amount of money.

“UnitTests.py” contains the unit tests. It contains functions named “test\_greedy” and “test\_dynamic” that tests the greedy and dynamic functions respectively.

Another file named “analysis.py” contains the code to find 10 numbers for which Dynamic programming solution is optimal. For this purpose, as discussed in lab, random numbers are generated and passed to both the functions. The number for which the number of coins required are less for dynamic then greedy algorithm, is actually the number for which dynamic solution is optimal. In general, we know that dynamic approach is more optimal than the greedy approach but for the given denominations, both greedy and dynamic approaches act in the same way.

# How to run:

It is a python code. Use any python IDE to run it, IDLE or Jupyter Notebook are recommended. I’ve implemented the functions, so the user just have to call them and pass them the amount whose change is required and a list of denominations.

# GitHub Link:

<https://github.com/13bscsmumar/Lab-8-Change-Making-Problem>