WAQAS RAZA  
5924  
BESE-4A

# Understanding

In this lab, we are supposed to implement famous Change-Making problem by using Greedy algorithm and Dynamic Programming. Goal is to choose fewest coins totaling N, where N is the amount of money to be changed entered by user.

# Implementation:

I have implemented the solution by reversing the given sorted array or first sort the array then reverse it since we want to get the max entry < amount entered by user without traversing the whole array. Keep on checking if the amount is larger than the available coins for change(an array) record that integer and subtract it from the amount. Stop when amount >= 0;

For Dynamic programming I first figured out min number of ways to make a change for a numbers available and then use those numbers to make a list of minimum entries to make a change for given number.

Here are some steps

if amount=0 then just return empty set to make the change, so 1 way to make the change.

if no coins given, 0 ways to change the amount.

Main thing To avoid these re-computations, we could store the results when computed and re-use them if required again. This reduces the time complexity of this algorithm to O(nm) where n is total amount to make change for and m is total number of denominations.ng in dynamic prog approach is memorization

# Github

https://github.com/waqasraza123/coin-changing-problem-in-java