
DSC 190 - Homework 05

Due: Wednesday, February 9

Write your solutions to the following problems by either typing them up or handwriting them on another piece of paper. Unless otherwise noted by the problem's instructions, show your work or provide some justification for your answer. Homeworks are due via Gradescope at 11:59 PM.

Problem 1.

Determine whether each of the following statements is True or False. If the statement is False, provide a counterexample.

- a) In the activity selection problem, assuming unique finish times, any optimal solution must contain the event with the earliest finish time.
- b) In the minimal spanning tree problem, assuming unique edge weights, any optimal solution must contain the graph edge whose weight is the smallest.

Programming Problem 1.

In a file named `make_change.py`, create a function `make_change(t)` which computes the number of ways of combining American quarters (worth 25 cents), dimes (10 cents), nickels (5 cents), and pennies (1 cent) to make t cents.

It may be possible that there is no way to make change for the target, t , in which case your function should return 0. You should assume that you are very rich and have an unlimited number of each type of coin.

For example: `make_change(11)` should be 3, since we can either use:

- 11 pennies,
- 1 nickel and 6 pennies, or
- 1 dime and 1 penny, or
- 2 nickels and 1 penny.

There is starter code on the course page, and you should submit your solution to the autograder.

Programming Problem 2.

Consider the same problem of computing the number of ways to make change, but now we are not rich – we only have a certain number of quarters, dimes, nickels, and pennies.

In `constrained_make_change.py`, write a function named `constrained_make_change(t, coins)`, where t is the target sum that we are trying to reach, and `coins` is a list of four elements containing the number of quarters, dimes, nickels, and pennies we have. The function should return the number of ways of adding up to t using only the coins we have.

For example, `constrained_make_change(25, [2, 0, 0, 25])` should return 2, since there are only two ways with the coins we have:

- 1 quarter, or
- 25 pennies.

There is starter code on the course page, and you should submit your solution to the autograder.