Activity Sheet 14

Manager name:

Recorder name:

Speaker name:

Section 8.1

1. We can approach the change-making problem by instead doing dynamic programming in two dimensions, j to represent that we use only the first j denominations D[1] through D[j], and n to represent the target value. So F(j, n) is the number of coins we need to use, being allowed to only use the first j denominations. (Notice that indexing in D starts at 1)

   a. Assuming we have the denominations D = [1, 4, 6], determine F(1, 5) as well as F(2,5).

   b. What should F(0, 0), F(2, 0), F(0, 1), F(2, −2) be? Your answers to this don't really depend on the particular denominators used, they apply broadly.

   c. We can build a recurrence relation for F(j, n) as follows: In our effort to make change for n using the j first denominations, we have two options: We can either use the D[j] denomination, and then we need to reach the target of n−d_j using the first j denominations still, or we can not use the D[j] denomination at all, meaning that we need to reach the target n using the first j−1 denominations. Use this statement to write a recurrence relation for F(j, n).

d. Write an algorithm that uses this recurrence relation to fill the 2-dimensional array F[0..m, 0..n]. Make sure to either avoid or handle carefully the case where you look up a value F[j, i] where i is a negative number.