CMPS 102 — Fall 2018 — Homework 3

*"I have read and agree to the collaboration policy." - **Kevin Wang***

Solution to Problem 4: Cakes

Given N dollars to spend in a shop that offers m varieties of cakes with distinct prices $S = \{S_1, \cdots, S_m\}$, find the possible combinations of cakes given that all money must be spent and that there are unlimited cakes of each variety.

Algorithm 1 Returns the number of cake combinations possible

```
**CAKE-COMBO** ($S[S_1, \cdots, S_m], m, N$):
Initialize $Table[0, \cdots, N][1, \cdots, m]$ // modified indexes
**for** $cost = 0$ and $variety = 1$ to $m$ **do**
    $Table[cost][variety] = 1$
**end for**
**for** $cost = 1$ to $N$ **do**
    **for** $variety = 1$ to $m$ **do**
        Let $c_i$ be the count of combinations including $S_{variety}$
        Let $c_e$ be the count of combinations excluding $S_{variety}$
        $Table[cost][variety] = c_i + c_e$
    **end for**
**end for**
$Table[N][m]$ contains the total count of combinations possible when spending $N$ dollars on a selection of $m$ varieties
```

The for-loops take time $O(m) + O(N) \cdot O(m)$. Thus the time complexity of this dynamic programming algorithm has time complexity: $O(mN)$.

The $Table$ takes $O(N) \cdot O(m)$. Thus the time complexity of this dynamic programming algorithm has space complexity: $O(mN)$.