Student: John Rizzo

Course: CS590-A Algorithms

Instructor: Dr. William Hendrix

Due Date: April 07, 2025

Description: Homework 5 Algorithms

Problem 1

An effective iteration order would be to enumerate from 1 to n in ascending order as, with memoization, the calculations would already have been computed and memoized.

```
for i in range(1, n+1):
    if i == 1:
        memo[i] = 1
    else:
        memo[i] = memo[i-1] + memo[i-2]
```

Problem 2

```
def MinSquares(n):
    # dp = new array of size (n + 1)
    dp[0] = 0 # base case
    for k from 1 to n:
        dp[k] = infinity
        x = 1
        while x^2 <= k:
            dp[k] = min(dp[k], 1 + dp[k - x^2])
            x += 1
    return dp[n]
```

Problem 3

I don't believe that the space complexity can be reduced as we are already minimizing the number of calculations which requires each square to be touched from 1 to n.

Problem 4

```
B[0] = 0
for i in range(1, n+1):
    B[i] = max { B[i-1], B[P[i]] + b_i}

# B[n] is the optimal solution
# printing in reverse gives us the optimal schedule
for i in range(len(B), 0, -1):
    print B[i]
```