Corey Savage
CS 325

Project 2

Algorithms:

A. Greedy Algorithm:

Pseudocode-

```
V[i];
While A > 0 {
    If V[i] <= A {
        A -= V[i]
        minCoins += 1
    }
    If V[i] > A {
        i -= 1
    }
}
```

Complexity - O(A)

B. Dynamic Programming:

Pseudocode-

```
S[V.size][ ]
S[0][0 .. A ] = (0)
T[ ] = S[0][0 ..A]
For i <= a {
    C = A
    While j < v.size and V[j] <= i {
        T = S[i - V[j]]
        T[j] += 1
        C2 += T[0...size]
        If C2 <= C {
            S = T
            C = C2
        }
        j += 1
    }
}
```

Complexity: O(AV)

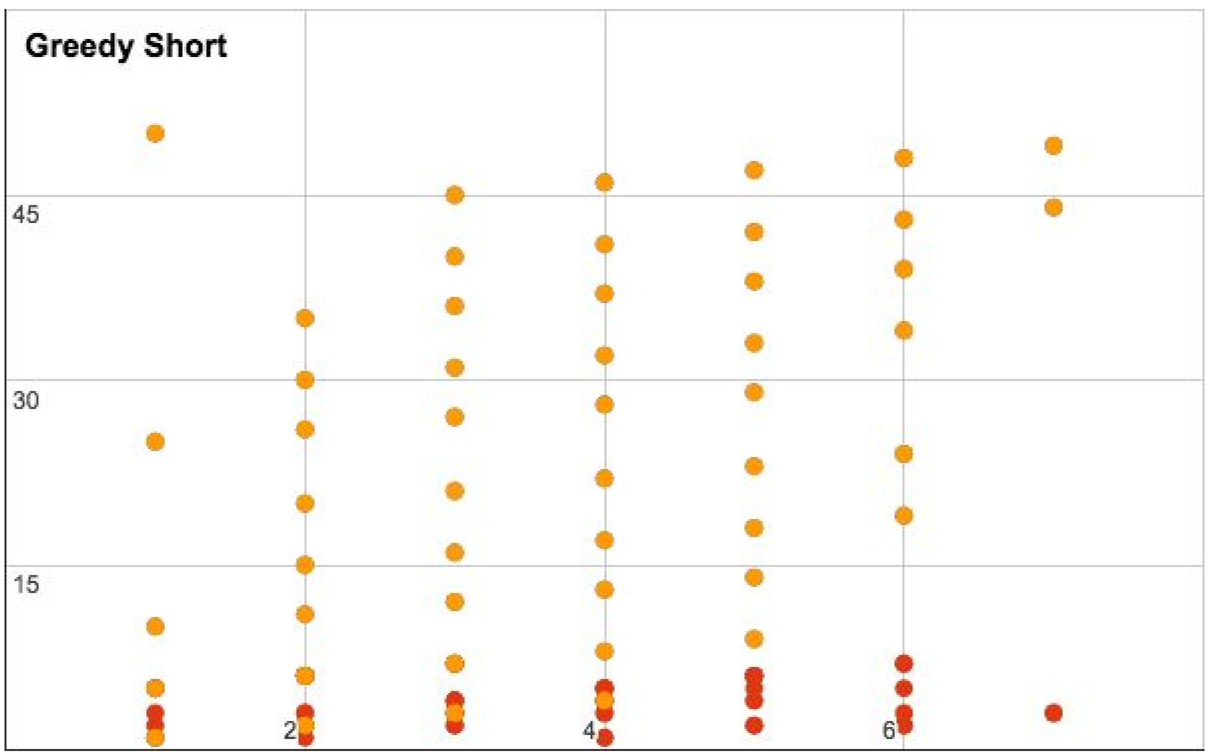
Table: The table for the Dynamic Programming algorithm stores the solutions in a 2d table for integers through A. Each A value beings by being assigned C = A, this is the worst possible solution which only works because all V[ ] sets must begin with 1. From there, the table will search previous solutions, and keeping the solution that returns the lowest C. Searching previous solutions is done by subtracting V[j] from the index of S.

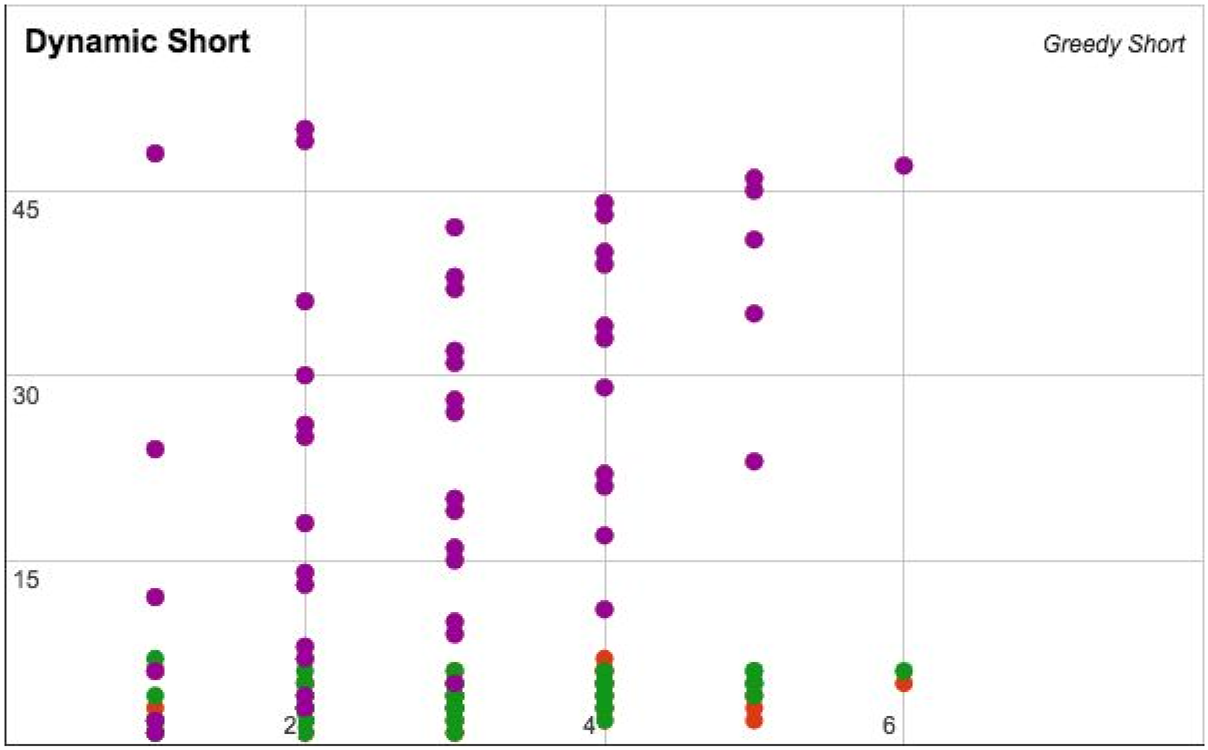
With each coin in V[j] used to find possible solutions, every possible one-coin-away solution is considered, which guarantees the optimal solution. This is a justifiable why to fill the table, since it minimizes the number of previous solution searches required while also guaranteeing the optimal change solution.

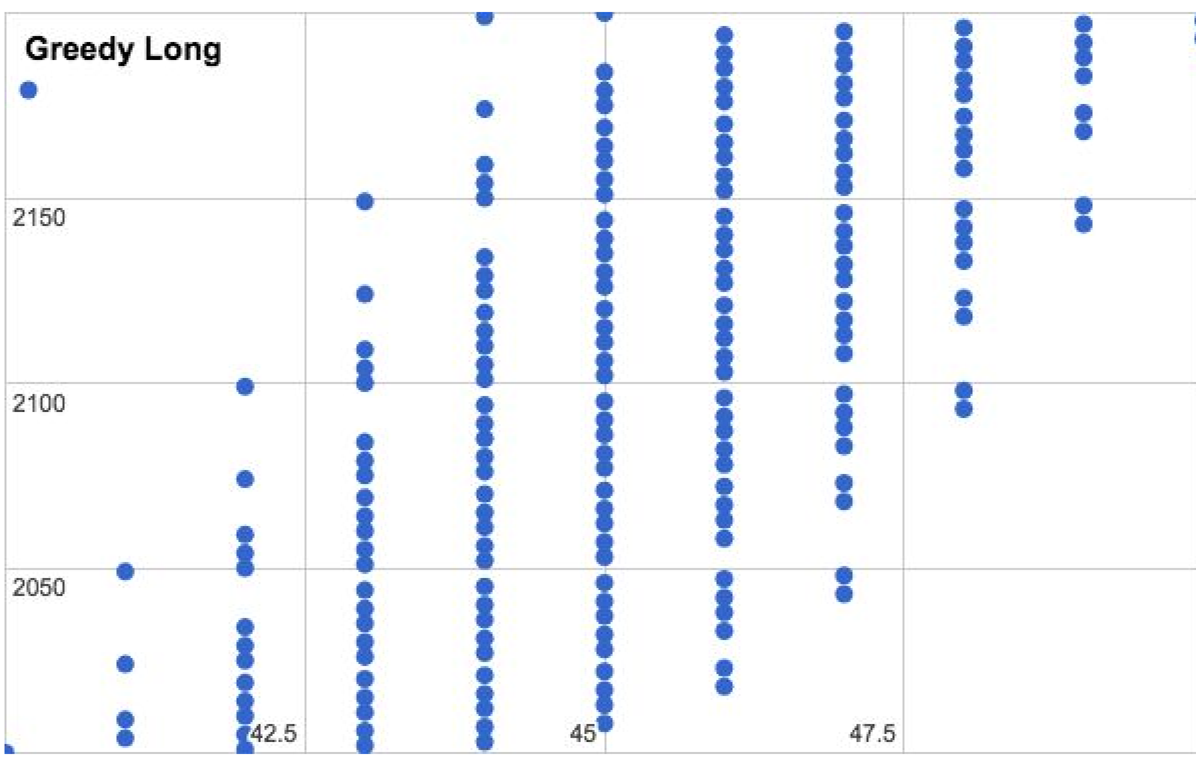
Test Results:

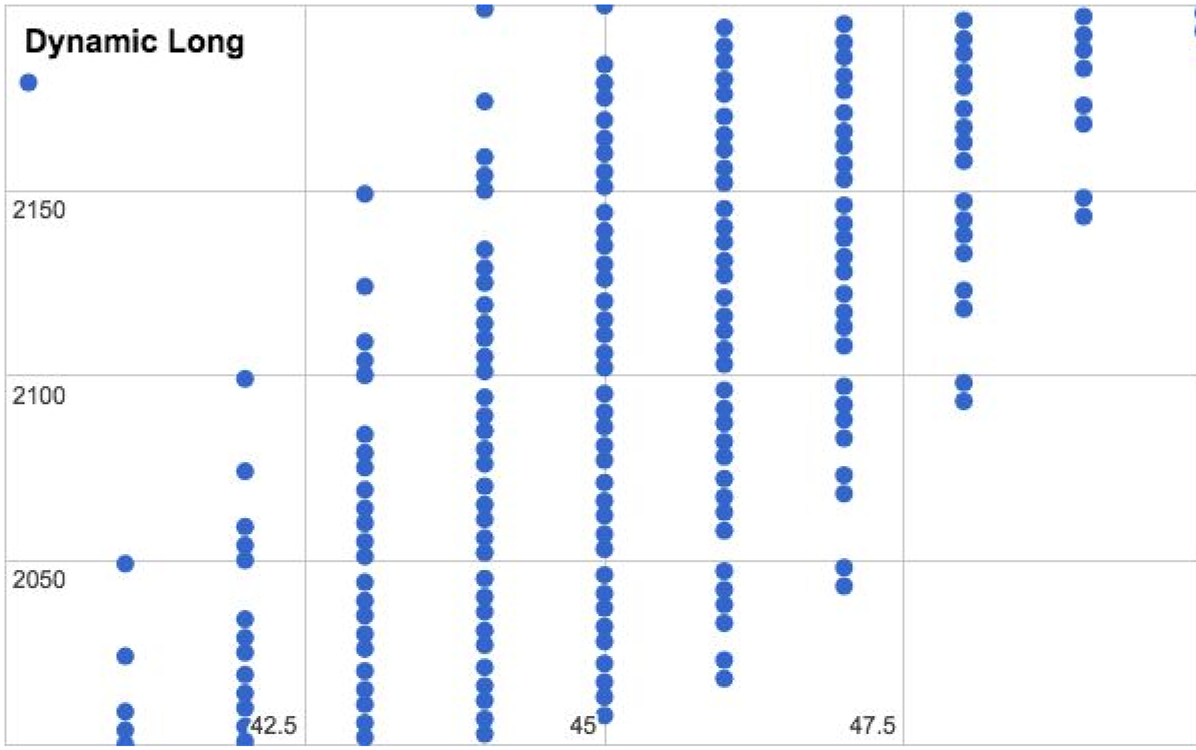
---

3. Coins:



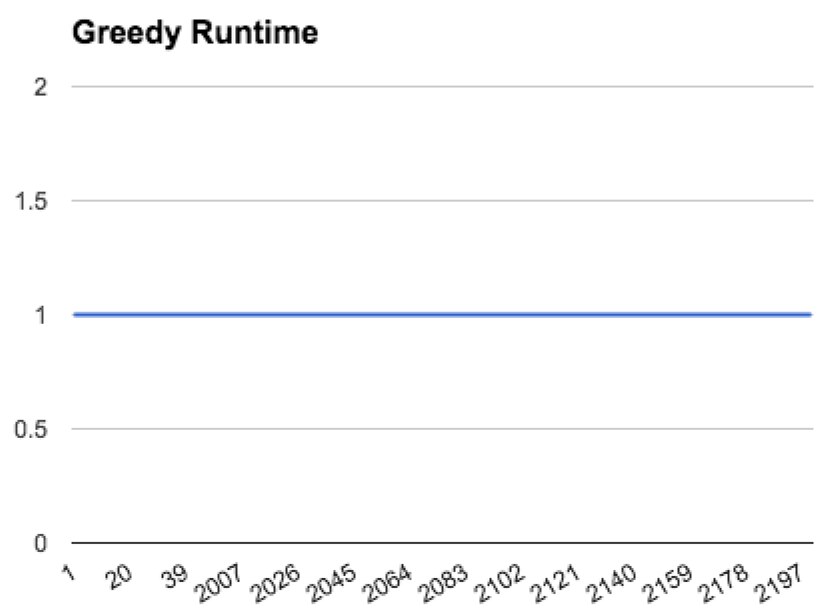

Dynamic Short
Greedy Short
45
30
15
2
4
6


Greedy Long
2150
2100
2050
42.5
45
47.5

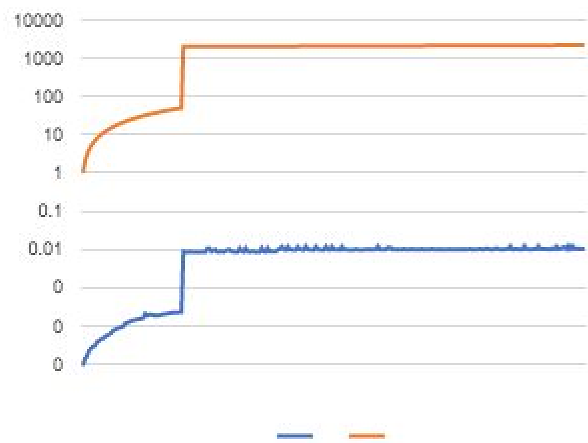


When looking at the long data 2000-2200 it seems like the growth of coin number is much more linear than when the coin values are lower.

4. Running Time:



5. Total Run Time:



6. Country with V = [1, 3, 9, 27]:
   The coin set given seems like it would be a canonical coin system. If the system is optimal for the greedy algorithm, it would make little sense to use the dynamic programming method. Since dynamic programming requires a substantially more complex algorithm, which as a result increases runtime. For all canonical coin systems, choosing greedy over dynamic benefits users runtime, with no drawbacks.

7. Optimal Greed Instances:
   V = {1, 5, 10, 25, 50}
   V= {1, 5, 10}
   V = {1,2,4,8}
   These coin sets produce optimal values because they are what's considered canonical coin systems. Canonical coin systems are sets where every possible solution is optimal for the greedy algorithm.