Dynamic Programming Algorithms

Source: Geeks for Geeks

- **Fibonacci numbers**
  - F[0] = 1
  - F[1] = 1
  - for n >=2, F[n] = F[n-1] + F[n-2]
- **Minimum Coin Change**
  - Given a value N, we want to make change for N with the minimum number of coins. Assume that we have an infinite supply of each of the S = {S1, S2, …, Sm} valued coins. What is the minimum number of coins needed to make the change? The order of the coins doesn't matter.
  - Example for N = 4 and S = {1, 2, 3}. Result = 2.
    - {1, 1, 1, 1}
    - {2, 1, 1}
    - {2, 2}
    - {3, 1}
  - Example for N = 10 and S = {2, 5, 3, 6}. Result = 2.
    - {2, 2, 2, 2, 2}
    - {2, 2, 6}
    - {2, 5, 3}
    - {2, 3, 5}
    - {5, 5}
- **The subset sum problem**
  - Given a set of non-negative integers, and a value "sum", find if there is a subset of the given set with sum equal to the given "sum".
  - If set = {3, 50, 2}, and sum = 5
  - Result is True, because subset (3, 2) has sum 5