Dynamic Programming
Coin Change – Minimum number of coins to Make Sum
Coins[1, 2, 5]. Sum = 11.

Coins/sum	0	1	2	3	4	5	6	7	8	9	10	11
	0											
	0											
	0											

Coins/sum	0	1	2	3	4	5	6	7	8	9	10	11
1	**0**	1	2	3	4	5	6	7	8	9	10	11
	0											
	0											

Coins/sum	0	1	2	3	4	5	6	7	8	9	10	11
1	**0**	1	2	3	4	5	6	7	8	9	10	11
2	**0**	1	1	2	2	3	3	4	4	5	5	6

Coins/sum	0	1	2	3	4	5	6	7	8	9	10	11
1	**0**	1	2	3	4	5	6	7	8	9	10	11
2	**0**	1	1	2	2	3	3	4	4	5	5	6
5	**0**	1	1	2	2	1	2	2	3	3	2	3

The minimum number of coins for a value sum can be computed using the recursive formula.

if $sum == 0$:

0 *coins required*

else if $sum > 0$:

$$minCoins(coins[0..m\text{-}1], sum) = min\{1 + minCoins(sum - coin[i])\}$$

$$where,\ 0 \le i \le m - 1\ and\ coins[i] \le sum.$$

Dynamic Programming

Coin Change – Minimum number of coins to Make Sum

Coins[1, 4, 5]. Sum = 12.

Coins/sum	0	1	2	3	4	5	6	7	8	9	10	11	12
	0												
	0												
	0												

Coins/sum	0	1	2	3	4	5	6	7	8	9	10	11	12
1	**0**	1	2	3	4	5	6	7	8	9	10	11	12

Coins/sum	0	1	2	3	4	5	6	7	8	9	10	11	12
1	**0**	1	2	3	4	5	6	7	8	9	10	11	12
4	**0**	1	2	3	1	2	3	4	2	3	4	5	3

Coins/sum	0	1	2	3	4	5	6	7	8	9	10	11	12
1	**0**	1	2	3	4	5	6	7	8	9	10	11	12
4	**0**	1	2	3	1	2	3	4	2	3	4	5	3
5	**0**	1	2	3	1	1	2	3	2	2	2	3	3