What is the recurrence relation for the coin-row problem?

The coin-row problem is a dynamic programming problem that involves choosing a subset of coins from a given row such that the sum of the chosen coins is maximized while adhering to a constraint that no two adjacent coins can be chosen. The recurrence relation for this problem can be defined as:

$F(i) = \max \{F(i-1), F(i-2) + v(i)\}$

Where F(i) is the maximum value that can be obtained from the first i coins, v(i) is the value of the i-th coin, and i is the index of the coin. The recurrence relation is defined for i = 1 to n, where n is the total number of coins.

The base case for this recurrence relation is F(0) = 0, which represents that no coins can be chosen if there are no coins in the row.

In this recurrence relation, F(i) is the maximum value that can be obtained from the first i coins. The maximum value can be obtained by either choosing the i-th coin or not choosing the i-th coin. If the i-th coin is chosen, then the maximum value is the value of the coin plus the maximum value that can be obtained from the first i-2 coins (to maintain the constraint that no two adjacent coins can be chosen). If the i-th coin is not chosen, then the maximum value is the maximum value that can be obtained from the first i-1 coins.

This recurrence relation can be solved using either a bottom-up or top-down approach. The bottom-up approach is done by filling a table, while the top-down approach is done by using recursion. Both approach have the same time complexity which is O(n)

In the end, the maximum value obtained from the coin-row problem can be obtained from F(n), which represents the maximum value that can be obtained from the first n coins.