

# Maximize House Robbing

## Description

Dynamic Programming is mainly an optimization over plain recursion. Wherever we see a recursive solution that has repeated calls for same inputs, we can optimize it using Dynamic Programming. The idea is to simply store the results of subproblems, so that we do not have to re-compute them when needed later. This simple optimization reduces time complexities from exponential to polynomial.

For the following problem, your goal is to get as much money as you can for robbing houses. Each house has a certain amount of money stashed in it, the only problem is that you cannot rob adjacent houses because of a connected security system that will contact the police if two houses next to each other are broken into.

Given a list of non-negative integers representing the amount of money of each house, determine the maximum amount of money you can rob tonight without alerting the police. Return the total value.

## Test Cases

Input: [2,7,9,3,1]

Output: 12

Input: [5,6,8,2,3,6,3,7,3,8,11,2,4,7,2,13,6,9,9,2]

Output: 68