

198. House Robber

<https://leetcode.com/problems/house-robber/description/>

```
class Solution {
public:
    int rob(vector<int>& nums) {
        int rob = 0;
        int norob = 0;
        for (int i = 0; i < nums.size(); i++) {
            int newRob = norob + nums[i];
            int newNoRob = max(norob, rob);
            rob = newRob;
            norob = newNoRob;
        }
        return max(rob, norob);
    }
};
```

The screenshot displays the LeetCode submission interface for the 'House Robber' problem. On the left, the C++ code is shown in a dark-themed editor. The code defines a class 'Solution' with a public method 'rob' that takes a vector of integers 'nums' and returns the maximum amount that can be robbed without alerting the police. The code uses a dynamic programming approach with two variables, 'rob' and 'norob', to track the maximum amount for each step. On the right, the submission results are shown. The status is 'Accepted', indicating that all 70 test cases passed. The runtime is 0 ms, which is 100.00% faster than the other submissions. The memory usage is 10.13 MB, which is 78.58% less than the other submissions. A bar chart shows the runtime distribution, with the submitted solution being the fastest. Below the results, the 'Testcase' tab is selected, showing the input 'nums = [1,2,3,1]' and the output '4'.

```
1 class Solution {
2 public:
3     int rob(vector<int>& nums) {
4         int rob = 0;
5         int norob = 0;
6         for (int i = 0; i < nums.size(); i++) {
7             int newRob = norob + nums[i];
8             int newNoRob = max(norob, rob);
9             rob = newRob;
10            norob = newNoRob;
11        }
12        return max(rob, norob);
13    }
14};
```

Accepted 70 / 70 testcases passed
lonethinker submitted at Mar 12, 2025 15:59

Runtime: 0 ms | Beats 100.00%
Memory: 10.13 MB | Beats 78.58%

Code: C++
class Solution {

Accepted Runtime: 0 ms
Case 1 Case 2

Input
nums =
[1,2,3,1]

Output
4

Submitted by,

Ankush Thakur | 22BCS11815

22BCS_IOT-637 (A)