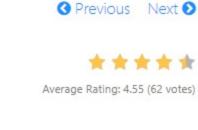
198. House Robber ^C

Articles > 198. House Robber ▼

March 5, 2016 | 69.4K views



6 💟 🛅

money stashed, the only constraint stopping you from robbing each of them is that adjacent houses have security system connected and it will automatically contact the police if two adjacent houses were broken into on the same night. Given a list of non-negative integers representing the amount of money of each house, determine the

You are a professional robber planning to rob houses along a street. Each house has a certain amount of

maximum amount of money you can rob tonight without alerting the police.

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

Example 1:

```
Output: 4
 Explanation: Rob house 1 (money = 1) and then rob house 3 (money = 3).
               Total amount you can rob = 1 + 3 = 4.
Example 2:
```

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

```
Output: 12
Explanation: Rob house 1 (money = 2), rob house 3 (money = 9) and rob house 5 (money =
             Total amount you can rob = 2 + 9 + 1 = 12.
```

• 0 <= nums[i] <= 400

Constraints:

• 0 <= nums.length <= 100

Approach #1 (Dynamic Programming) [Accepted]

Solution

It could be overwhelming thinking of all possibilities on which houses to rob.

A natural way to approach this problem is to work on the simplest case first.

Algorithm

Let us denote that:

f(k) = Largest amount that you can rob from the first k houses.

Now, let us look at n = 2, which $f(2) = \max(A_1, A_2)$.

Let us look at the case n = 1, clearly $f(1) = A_1$.

 A_i = Amount of money at the ith house.

1. Rob the third house, and add its amount to the first house's amount.

 $f(k) = \max(f(k-2) + A_k, f(k-1))$

public int rob(int[] num) {

Rate this article: * * * * *

O Previous

int temp = currMax;

For n = 3, you have basically the following two options:

Clearly, you would want to choose the larger of the two options at each step.

Therefore, we could summarize the formula as following:

2. Do not rob the third house, and stick with the maximum amount of the first two houses.

We choose the base case as f(-1) = f(0) = 0, which will greatly simplify our code as you can see.

step you only need the previous two maximum values, two variables are suffice.

currMax = Math.max(prevMax + x, currMax);

int prevMax = 0; int currMax = 0; for (int x : num) {

The answer will be calculated as f(n). We could use an array to store and calculate the result, but since at each

```
prevMax = temp;
      }
      return currMax;
  }
Complexity analysis
   • Time complexity : O(n). Assume that n is the number of houses, the time complexity is O(n).
   • Space complexity : O(1).
```

Next

```
Comments: 23
                                                                                                       Sort By -
              Type comment here... (Markdown is supported)
              Preview
                                                                                                         Post
              jjjj00000 ★97 ② December 15, 2019 9:42 PM
              This problem is at easy level? seriously?
              97 A V C Share  AReply
              SHOW 11 REPLIES
              NideeshT ★ 581 ② April 23, 2019 10:49 AM
              (Java) Whiteboard Youtube Video Explanation - accepted
              https://www.youtube.com/watch?v=gQzNp8gBn5k&lc=
                                                         Read More
              29 A V C Share  Reply
              SHOW 2 REPLIES
              numberMumbler # 25 October 31, 2017 10:55 PM
              tdhaayushverma For this particular problem, you could use negatives (robber loses money by visiting
              the house). The optimal strategy would be to skip any house with a negative value... so line 6 would
              become something like this:
              currMax = Math.max(prevMax + Math.max(0, x), currMax);
              15 A V Share  Reply
              SHOW 1 REPLY
              Oldman09 * 82 October 5, 2016 10:22 AM
              The crucial part of this dynamic programming question is the granular case includes n=3.
              12 A V C Share  Reply
              Isheng_mel ★ 169 ② October 3, 2019 7:02 PM
              For n = 3, you have basically the following two options:
              1. Rob the third house, and add its amount to the first house's amount.
                                                        Read More
              5 A V C Share   Reply
              SHOW 3 REPLIES
              xuhanreconcil *5 O August 9, 2018 7:13 AM
              I don't get it why f(k) = max(f(k-2) + Ak, f(k-1)) works. f(k-1) is largest amount can rob from 0 to k-
              1, but it may or may not need to rob k-1 house. If not rob k-1 house, then f(k - 1) + Ak also needs take
              into consideration.
              What I can only understand is see f(k) as largest amount you can rob from first k house and you have to
                                                         Read More
              5 A V C Share   Reply
              SHOW 1 REPLY
              yingxuegu 🛊 3 🖸 September 22, 2017 7:10 AM
              The crucial part of this dynamic programming question is the granular case includes n=3.
              good idea, from Oldman09
              3 A V C Share   Reply
              mission_2020 # 98 @ April 26, 2020 9:17 PM
              change the complexity to medium please
              2 A V C Share   Reply
              al_lea 🛊 3 🧿 August 3, 2019 2:52 AM
              @xuhanreconcil:
```

This is actually a little more mind burning than expected if you think thru. My thought process is this

Read More

this one shouldn't be tagged easy, while it's an easy DP, it is definitely harder than most easy problems

need to involve the f(k-3), but in the end the formula:

3 A V C Share Reply

like sum ints or two sum etc.

(1 2 3)

1 A V C Share Share

Username1604 * 41 • June 1, 2020 1:25 AM