**198. House Robber**

Easy

4846149Add to ListShare

You are a professional robber planning to rob houses along a street. Each house has a certain amount of 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 maximum amount of money you can rob tonight **without alerting the police**.

**Example 1:**

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

**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 = 1).

  Total amount you can rob = 2 + 9 + 1 = 12.

小偷偷钱，连续偷两家就会报警，怎么最大

他是符合最优子结构的，比如说前n项，一定有个偷法收益最高

符合问题存在边界

毫无疑问动态规划

动态规划就是要设一个dp[]

第一项为0

第二项等于原第一项

如果这一家+前两家大于上一家的最优解，那么就选这一家+前两家，要么就等于上一家也就是这家不偷

最后动态规划总是return最后一个

class Solution {

public int rob(int[] nums) {

if (nums.length==0) return 0;

int[] dp=new int[nums.length+1];

dp[0]=0;

dp[1]=nums[0];

for(int i=1;i<dp.length-1;i++){

dp[i+1]=Math.max(dp[i],dp[i-1]+nums[i]);

}

return dp[dp.length-1];

}

}