198. House Robber

Easy,
Dynamic Programming.

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: [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: [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.
```

解法

该题是比较经典的动态规划的题。
只要梳理好状态转换方程，即可解出。

该题的状态转换方程是，
偷一家或者偷上上一家+当前这家中 选大的那一个。
所以是， dp[i]=Math.max(dp[i-1],dp[i-2]+nums[i])。

Java

```java
class Solution {
    public int rob(int[] nums) {
        int len = nums.length;
        int[] dp = new int[len+2];
        for(int i=0; i<=len;i++){
            dp[i]=0;
        }
        for(int i=1; i<len+1;i++){
            dp[i+1]=Math.max(dp[i],dp[i-1]+nums[i-1]);
        }
        return dp[len+1];
    }
}
```