198. House Robber

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 systems connected and **it will automatically contact the police if two adjacent houses were broken into on the same night**.

Given an integer array `nums` representing the amount of money of each house, return *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.

```
def rob(self, arr: List[int]) -> int:
        n = len(arr)
        if len(arr)<3:
            return max(arr)
        dp = [0]*(n+1)
        dp[0] = 0
        dp[1] = arr[0]
        dp[2] = max(arr[0],arr[1])
        for i in range(3,n+1):
            dp[i] = max(arr[i-1]+max(dp[i-2],dp[i-3]),dp[i-1])
        return dp[n]
```

```
def rob(self, nums: List[int]) -> int:
        if len(nums)==1:
            return nums[0]
        if len(nums)==2:
            return max(nums)
        dp = [0]*len(nums)
```

```
        dp[0] = nums[0]
        dp[1] = max(nums[1],nums[0])
        for i in range(2,len(nums)):
            dp[i] = max(dp[i-1],dp[i-2]+nums[i])
        return dp[-1]
```