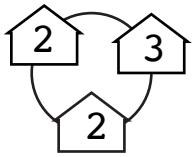
213. House Robber II
Medium

You are a professional robber planning to rob houses along a street. Each house has a certain amount of money stashed. All houses at this place are arranged in a circle. That means the first house is the neighbor of the last one. Meanwhile, adjacent houses have a security system connected, and it will automatically contact the police if two adjacent houses were broken into on the same night.

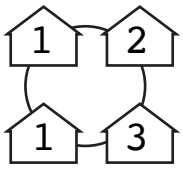
Given an integer array nums representing the amount of money of each house, return the maximum amount of money you can rob tonight without alerting the police.

Example 1:



Input: nums = [2,3,2]
Output: 3
Explanation: You cannot rob house 1 (money = 2) and then rob house 3 (money = 2), because they are adjacent houses.

Example 2:



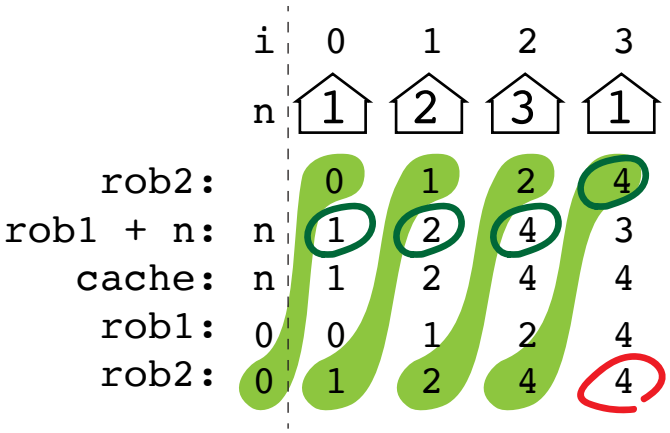
Input: nums = [1,2,3,1]
Output: 4
Explanation: Rob house 1 (money = 1) and then rob house 3 (money = 3).
Total amount you can rob = 1 + 3 = 4.

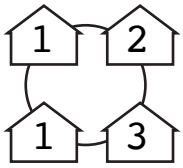
Example 3:

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

```python
class Solution:
    def rob(self, nums: List[int]) -> int:
        return max(nums[0], self.analysis(nums[1:]), self.analysis(nums[:-1]))

    def analysis(self, nums):
        rob1, rob2 = 0, 0
        for n in nums:
            cache = max(rob1 + n, rob2)
            rob1 = rob2
            rob2 = cache
        return rob2
```






```
i:0 1 2 3
nums[0]   = [1]
nums      = [1,2,3,1]
nums[1:]  =   [2,3,1]
nums[:-1] = [1,2,3]
```

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

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

perform the 'house robber' algorithm on two parts of the nums list, that either (1) excludes the first number, or (2) excludes the last number