

당신은 전문털이범이다. 어느 집에서든 돈을 훔쳐올 수 있지만 경보 시스템 때문에 바로 옆집은 훔칠 수 없고 한 칸 이상 떨어진 집만 가능하다. 각 집에는 훔칠 수 있는 돈의 액수가 입력값으로 표기되어 있다. 훔칠 수 있는 가장 큰 금액을 출력하라.

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.

### 1. 브루트 포스 재귀

class Solution:

```
def rob(self, nums: List[int]) -> int:
    def _rob(i):
        if i < 0:
            return 0
        return max(_rob(i - 1), _rob(i - 2) + nums[i])

    return _rob(len(nums) - 1)
```

▸ 시간 초과

### 2. 다이나믹 프로그래밍

class Solution:

dp = {}

```
def rob(self, nums: List[int]) -> int:
    def _rob(i):
        if i < 0:
            return 0
        elif i in self.dp:
            return self.dp[i]
        else:
            self.dp[i] = max(_rob(i - 1), _rob(i - 2) + nums[i])
            return self.dp[i]

    return _rob(len(nums) - 1)
```

▸ 타블레이션(하향식)

import collections

class Solution:

```
def rob(self, nums: List[int]) -> int:
    if not nums:
        return 0
    if len(nums) <= 2:
        return max(nums)
    dp = collections.OrderedDict()
```

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

- 타블레이션(하향식)