Problem 213: House Robber II

Problem Information

Difficulty: Medium
Acceptance Rate: 44.27%
Paid Only: No
Tags: Array, Dynamic Programming

Problem Description

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

Constraints:

* `1 <= nums.length <= 100` * `0 <= nums[i] <= 1000`

Code Snippets

C++:

```
class Solution {
public:
int rob(vector<int>& nums) {

}
};
```

Java:

```
class Solution {
public int rob(int[] nums) {

}
}
```

Python3:

```
class Solution:
def rob(self, nums: List[int]) -> int:
```