```java
/*Author: Bochen (mddboc@foxmail.com)
Last Modified: Tue Apr 10 22:28:44 CST 2018*/

/*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.*/

import java.util.*;


class TreeNode {
    int val;
    TreeNode left;
    TreeNode right;

    TreeNode(int x) {
        val = x;
    }
}

public class Test {
    public static void main(String[] args) {

        int num = 2147483648;

        new Solution().reverseBits(num);
    }
}


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

        if (nums == null || nums.length == 0) {
            return 0;
        } else if (nums.length == 1) {
            return nums[0];
        } else if (nums.length == 2) {
            return Math.max(nums[0], nums[1]);
        }

        int numsLength = nums.length;

        int[] storage = new int[numsLength];
        storage[0] = nums[0];
        storage[1] = Math.max(nums[0], nums[1]);

        for (int i = 2; i < numsLength; i++) {

            int temp = storage[i - 2] + nums[i];
            storage[i] = Math.max(temp, storage[i - 1]);
        }

        return storage[numsLength - 1];
    }
}
```