**1.12 MAXIMUM AMOUNT OF MONEY A ROBBER CAN ROB**

**AIM**:

To find the maximum amount of money a robber can rob from houses arranged in a circle, such that no two adjacent houses are robbed.

**ALGORITHM:**

1. Observation:

* Because houses are in a circle, the first and last house cannot both be robbed.

• So, break the problem into two cases:

• Case 1: Rob houses from 0 to n-2 (exclude last house).

• Case 2: Rob houses from 1 to n-1 (exclude first house).

• The answer = max(Case 1, Case 2).

2. Subproblem (House Robber I):

• For a linear arrangement of houses, use dynamic programming:

• dp[i] = max(dp[i-1], dp[i-2] + nums[i])

• Base cases:

• dp[0] = nums[0]

• dp[1] = max(nums[0], nums[1])

3. Handle edge cases:

• If nums is empty → return 0.

• If nums has only 1 element → return nums[0].

**PROGRAM**:

A screenshot of a computer program

AI-generated content may be incorrect.

Input:

nums = [1, 2, 3, 1]

Output:

A screenshot of a computer

AI-generated content may be incorrect.

**RESULT:**

Thus the program is successfully executed, and the output is verified.

**PERFORMANCE ANALYSIS:**

• Time Complexity: O(n) (one DP pass for each case).

• Space Complexity: O(1) (if optimized to use just two variables).