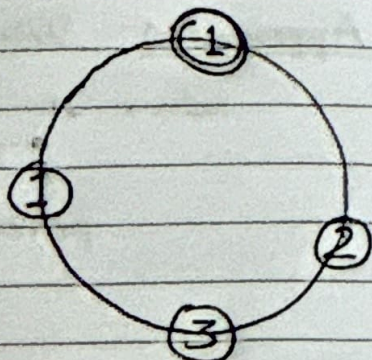


LEETCODE | 213 |

House Robber II

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

Output: 4



The difference from previous problem i.e. Leetcode 198 - House Robber I is the houses are in circle so, the robber cannot rob 0th & (n-1)th index on the same day

We can follow the previous approaches with a small twist.

Step 1 Divide nums into two arrays such that

nums: [1, 2, 3, 1]

nums1 → [1, 2, 3]

nums2 = [2, 3, 1]

int n = nums.length;

Step 2 int[] dp1 = new int[n-1];

int[] dp2 = new int[n-1];

Arrays.fill(dp1, -1);

Arrays.fill(dp2, -1);

Step 3 Return → Math.max(getMax(nums1, dp1), getMax(nums2, dp2));