Question: https://leetcode.com/problems/house-robber/

So the decision is simple we rob the present house if the sum of robbed mony from present house and present-2 position house is greater than the till now robbed money else we don’t rob the present house.

Code:  
class Solution {

public int rob(int[] nums) {

if(nums.length==0) return 0;

if(nums.length==1) return nums[0];

//Initialize an arrays to store the money

int[] mark = new int[nums.length];

//We can infer the formula from problem:mark[i]=max(num[i]+mark[i-2],mark[i-1])

//so initialize two nums at first.

mark[0] = nums[0];

mark[1] = Math.max(nums[0], nums[1]);

//we store maximum that can be obtained at that position

//Using Dynamic Programming to mark the max money in loop.

for(int i=2;i<nums.length;i++){

mark[i] = Math.max(nums[i]+mark[i-2], mark[i-1]);

}

return mark[nums.length-1];

}

}

Github Link :<https://lnkd.in/ecwtJeaz>