1. House Robber II

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 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**.

**Example 1:**

Input: [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: [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.

**解** 第一家和最后一家不能同时打劫，因此分别去掉第一家和最后一家，计算能够抢到的最大值

class Solution {  
public:  
 int rob(vector<int>& nums) {  
 if(nums.size() == 0)return 0;  
 if(nums.size() == 1)return nums[0];  
 if(nums.size() == 2)return max(nums[0], nums[1]);  
 return max(rob(nums, 0, nums.size() - 1),rob(nums, 1, nums.size()));  
 }  
 int rob(vector<int> &nums, int left, int right){  
 vector<int>dp(nums.size(), 0);  
 dp[left] = nums[left];  
 dp[left+1] = max(nums[left], nums[left + 1]);  
 for(int i = left + 2; i < right; ++i){  
 dp[i] = max(dp[i-1], dp[i-2] + nums[i]);  
 }  
 return dp[right-1];  
 }  
};