House Robber

由题意这是一个动态规划问题（dp问题），分析可设：

dp[i][0] 表示第i个房子没有偷的最大值

dp[i][1] 表示第i个房子偷了的最大值

所以有：

dp[i][0]=max(dp[i-1][0],dp[i-1][1])

dp[i][1]=dp[i-1][0]+nums[i]

初始值为：dp[0][0]=0 dp[0][1]=nums[0]

代码如下：

class Solution {

public:

int rob(vector<int>& nums) {

int size;

size=nums.size();

if(size==0)

return 0;

int dp[size][2];

dp[0][0]=0;

dp[0][1]=nums[0];

for(int i=1;i<size;i++)

{

dp[i][0]=max(dp[i-1][0],dp[i-1][1]);

dp[i][1]=dp[i-1][0]+nums[i];

}

return max(dp[size-1][0],dp[size-1][1]);

}

};