**动态规划：**

1. **确定每一个位置的状态是什么**
2. **确定状态转移方程**
3. **初始化**
4. **是否可以优化**

|  |
| --- |
| 状态：当前房屋选择偷还是不偷，不偷，他的利润为前一房屋的利润，偷则为第前两间的加当前的  状态转移方程：Math.max(dp[i - 1],dp[i - 1] + nums[i]);  初始化：dp[0] = 0, dp[1] = nums[0];  考虑是否可以优化：能（怎么优化？？？）      环形房屋：将其拆解，分为两种方式偷   1. 从0 ~ len - 1偷 2. 从1 ~ len偷         状态：经过当前位置时其最小值是多少  状态转移方程：grid[i][j] = Math.min( grid[i - 1][j], grid[i][j - 1]) + grid[i][j];      状态：经过当前位置时有多少条路径  状态转移方程：dp[i][j] = dp[ i - 1][j] + dp[i][j - 1]; |