House Robber

You are a professional robber planning to rob houses along a street. Each house has a certain amount of money stashed, the only constraint stopping you from robbing each of them is that 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**.

Credits:
Special thanks to @ifanchu for adding this problem and creating all test cases. Also thanks to @ts for adding additional test cases.

Solution 1

```
#define max(a, b) ((a)>(b)?(a):(b))
int rob(int num[], int n) {
    int a = 0;
    int b = 0;

    for (int i=0; i<n; i++)
    {
        if (i%2==0)
        {
            a = max(a+num[i], b);
        }
        else
        {
            b = max(a, b+num[i]);
        }
    }

    return max(a, b);
}
```

written by 452750465%40qq.com original link here

Solution 2

```
public int rob(int[] num) {
    int[][] dp = new int[num.length + 1][2];
    for (int i = 1; i <= num.length; i++) {
        dp[i][0] = Math.max(dp[i - 1][0], dp[i - 1][1]);
        dp[i][1] = num[i - 1] + dp[i - 1][0];
    }
    return Math.max(dp[num.length][0], dp[num.length][1]);
}
```

dp[i][1] means we rob the current house and dp[i][0] means we don't,

so it is easy to convert this to O(1) space

```
public int rob(int[] num) {
    int prevNo = 0;
    int prevYes = 0;
    for (int n : num) {
        int temp = prevNo;
        prevNo = Math.max(prevNo, prevYes);
        prevYes = n + temp;
    }
    return Math.max(prevNo, prevYes);
}
```

written by tusizi original link here

Solution 3

```
public int rob(int[] num) {
    int rob = 0; //max monney can get if rob current house
    int notrob = 0; //max money can get if not rob current house
    for(int i=0; i<num.length; i++) {
        int currob = notrob + num[i]; //if rob current value, previous house must
not be robbed
        notrob = Math.max(notrob, rob); //if not rob ith house, take the max valu
e of robbed (i-1)th house and not rob (i-1)th house
        rob = currob;
    }
    return Math.max(rob, notrob);
}
```

written by wenwen original link here