动态规划：413. 等差数列划分

主要思想：

状态表示：dp[i] 表示以i位置为结尾所有子数组中等差数列的个事

状态转移方程 1. 如果 c-b = b-a dp[i] = dp[i-1] + 1

2.如果c-b != b-a dp[i] = 0;

代码：

```
class Solution {
public:
    int numberOfArithmeticSlices(vector<int>& nums) {
        int n = nums.size();
        int sum = 0;
        vector<int> dp(n);
        for(int i = 2;i < n;i++)
        {
            if(nums[i]- nums[i-1] == nums[i-1]-nums[i-2])
            {
                dp[i] = dp[i-1]+1;
            }
            else
            {
                dp[i] = 0;
            }
        }
        for(auto e : dp)
        {
            sum += e;
        }
        return sum;
    }
};
```

动态规划：978. 最长湍流子数组

主要思想：

状态表示：1.f[i] 表示最后一个为“上升”的最长湍流子数组

2.g[i] 表示最后一个为“下降”的最长湍流子数组

状态转移方程：

1.如果num < num-1 g[i] = f[i-1] + 1

2. num > num-1 f[i] = g[i-1] + 1

代码：

```
class Solution {
public:
    int maxTurbulenceSize(vector<int>& arr) {
        int n = arr.size();
        vector<int> f(n,1);
        auto g = f;
        int ret = 1;
        for(int i = 1; i < n;i++)
        {
            if(arr[i] > arr[i-1])
            {
                f[i] = g[i-1] + 1;

            }
            else if(arr[i] < arr[i-1])
            {
                g[i] = f[i-1] + 1;
            }
            ret = max(ret,max(f[i],g[i]));
        }
        return ret;
    }
};
```