

410.分割数组的最大值

题目大意

对一个非负整数数组 $nums$ 切 $k - 1$ 刀，求这 k 部分各自和的最大值的最小值。

题目分析

乍一看又是“最大值”又是“最小值”似乎十分拗口，实际上他们指的是两个完全不同的过程，对我们切出来的这 k 部分，我们求一遍它们和的最大值，选取一种最巧妙的切割方式，最小化这个最大值。

首先观察一眼数据范围， $n = 1000$ ，感觉 $O(n^2)$ 的DP可解，于是开始考虑如何分解子问题。观察到每砍一刀产生的当前结果只与先前分割产生的整体结果有关，定义：

$dp[i][j]$ ：已经砍了 i 刀，到第 j 个元素为止，各块最大和的最小值

可以推导出状态转移方程：

$$dp[i][j] = \min_{0 \leq p < j} \max\{dp[i-1][p], \sum_{t=p+1}^j nums[t]\}$$

可通过提前处理出数组 $nums$ 的前缀和来简化复杂度。在砍了 0 刀时，各块最大和应为前缀和本身，从而边界条件为：

$$dp[0][i] = \sum_{t=0}^i nums[t]$$

由于 $dp[i]$ 要用到 $dp[i-1]$ 的结果，于是我们先枚举刀数 i ，然后枚举当前位置 j ，再需要一层循环枚举上一次切割位置 p ，最终输出的是 $dp[i-1][n-1]$ ，复杂度为 $O(kn^2)$ ，AC代码如下

```
int splitArray(vector<int>& nums, int k) {
    int sum[1005], n=nums.size();
    sum[0]=nums[0];
    for(int i=1; i<nums.size(); i++) sum[i]=sum[i-1]+nums[i];
    int minf[55][1005]; // i split, j elements;
    for(int i=0; i<n; i++) minf[0][i]=sum[i];
    for(int i=1; i<=k-1; i++){
        for(int j=0; j<n; j++){
            int minimum=1000000000;
            for(int p=0; p<j; p++){
                minimum=min(minimum, max(minf[i-1][p], sum[j]-sum[p]));
            }
        }
    }
    return minf[k-1][n-1];
}
```

```
        }
        minf[i][j]=mininum;
    }
}
return minf[k-1][n-1];
}
```