

一、题目说明

题目312. Burst Balloons, 有n个气球编号 $0 \sim n-1$, 每个气球上有一个数字, 存在数组nums中。每当戳破一个气球i的时候, 可以获得 $\text{nums}[\text{left}] * \text{nums}[i] * \text{nums}[\text{right}]$ 个硬币。求能获得的最大硬币数。难度是Hard!

二、我的解答

这个题目, 没解答出来。

先假设第一个被戳爆的气球为x, 则x两边的气球则产生了依赖; 那我们假设不戳爆x, 则x两边的气球就没有了依赖关系! 这个气球x, 我们可以放在最后戳爆它。

用dp解决, 状态转移方程: $\text{dp}[i][j] = \max(\text{dp}[i][j], \text{dp}[i][k] + \text{dp}[k][j] + \text{nums}[i] * \text{nums}[k] * \text{nums}[j])$, 其中k表示区间[i,j]中最后戳破的一个气球。dp[i][j] 表示第i至第j个元素这个区间能获得的最大硬币数。

```
class Solution {
public:
    int maxCoins(vector<int>& nums) {
        //nums[-1]=nums[n]=1
        nums.insert(nums.begin(),1);
        nums.push_back(1);

        int n=nums.size();

        int dp[n][n]; //dp[i][j]表示戳破第i至第j个元素这个区间能获得的最大硬币数
        for(int i=0;i<n;i++)
            for(int j=0;j<n;j++)
                dp[i][j]=0;

        for(int r=2;r<n;r++) //r为区间长度
            for(int i=0;i<n-r;i++){ //i为左区间
                int j=i+r; //j为右区间
                for(int k=i+1;k<j;k++)
                    dp[i][j]=max(dp[i][j],dp[i][k]+dp[k][j]+nums[i]*nums[k]*nums[j]);
            }

        return dp[0][n-1];
    }
};
```

性能如下:

```
Runtime: 12 ms, faster than 99.41% of C++ online submissions for Burst Balloons.
Memory Usage: 8.8 MB, less than 90.00% of C++ online submissions for Burst Balloons.
```

三、优化措施

无, 还要继续努力!