

FOI夏令营基础算法1 day8 解题报告

T1: 涂色 color

本题是一维线性差分模板题，只要理解差分思想，可以秒AC的题目。

T2 : 数字游戏 game

本题是一道需要前缀和预处理的区间DP经典题目。由于全程是对静态区间求最优值，用前缀和维护区间和最合适，但是区间和不能维护求模结果，因此预处理分两部分组成，最终求出区间k值。

区间DP部分：作为前置知识的复习巩固，一般从结果出发设计DP状态 $F[x][y][k]$ 表示区间 x 到 y 分为 k 个区间的最值。

得到状态转移方程： $F[x][y][k] = \max F[x][y][k], F[x][h-1][k-1] * K_{h,r}$, h 为第 k 部分开始位置， r 为区间末尾。

边界条件： $F[x][y][1] = K_{x,y}$

核心代码1：前缀和预处理

```
for(int i=1;i<=n;i++){
    a[i]=read();
    a[n+i]=a[i]; //模拟成环
}
//预处理1
for(int i=1;i<=2*n;i++){
    s[i]=s[i-1]+a[i];
}
//预处理2
for(int i=1;i<=n;i++){ //区间长度
    for(int j=1;j+i-1<=2*n;j++){
        int r=j+i-1; //结束位置
        su[j][r]=mod(s[r]-s[j-1]);
    }
}
```

核心代码2：区间DP

```
memset(dpmin,0x3f,sizeof(dpmin));
for(int i=1;i<=n;i++){
    for(int j=1;j<=n;j++){
        int r=j+i-1;
        dpmax[j][r][1]=dpmin[j][r][1]=su[j][r];
        for(int k=2;k<=m&&k<=i;k++){
            for(int h=j+k-1;h<=r;h++){
                dpmax[j][r][k]=max(dpmax[j][r][k],dpmax[j][h-1][k-1]*su[h][r]);
                dpmin[j][r][k]=min(dpmin[j][r][k],dpmin[j][h-1][k-1]*su[h][r]);
            }
        }
    }
}
```

T3: 求差分矩阵

本题是一道直接求解差分矩阵的快捷方法，因为直接用数学推导公式太繁琐。因此我们可以将差分矩阵中的每一个元素一个个的插进去。

```
void insert(int x1,int y1,int x2, int y2, int d){
    b[x1][y1]+=d;
    b[x2+1][y1]-=d;
    b[x1][y2+1]-=d;
    b[x2+1][y2+1]+=d;
}
```

因此，我们可以先假设原矩阵全是0，因此差分矩阵也全为0，接下来依次更新矩阵中每个元素的实际值。顺带计算差分矩阵。

T4: 牧场行走

本题是可以直接用朴素LCA求解，需要注意的是递归寻找父亲节点的过程中需要顺带记录与父亲节点的距离。然后多次询问的两点距离就是询问区间端点和各自LCA的距离和。