重庆八中周赛Round#27

时间：2024年3月15日

T1 放学回家 back (AcWing 3268)

模拟题。 题目的核心在于求遇到交通的时刻，交通灯处于什么样的状态，以及根据交通灯的状态得出需要等待的时间。

1. 红灯（r）：等到绿灯即可通行
2. 绿灯（g）：直接通行
3. 黄灯（y）：**黄灯等完后还要等到红灯结束才能通行**

需要明确，交通灯的状态是循环的，而循环的周期就是$(r+y+g)$，所以求解时可以将经过的时间段对周期取模。

例如$k=1$，当前是红灯，根据当前交通灯剩余的时间t将分成区间$[0,t),[t,t+g),[t+g,t+g+y),[t+g+y,g+y+r)$，看取余得到的时间落在哪个区间就可以知道交通灯状态了。

答案需要用`long long`存储。

T2 任务分配 matching (Atcoder dp O)

我们令$\mathrm{dp}[i][status]$表示安排了前i个人，这i个人占用的**任务**是$status$的时候的方案数

初始情况很简单，我们只需要简单的将$\mathrm{dp}[0][i]=mp[0][i]$即可

随后，对于某个状态$\mathrm{dp}[i][status]$，我们只需要枚举我们的第i个人选择了哪样任务，记为j，然后，我们将$\mathrm{dp}[i-1][status \bigoplus (1<<j)]$加到我们的$\mathrm{dp}[i][status]$即可

这样，我们的总时间复杂度为$O(n^2 2^n)$

注意到，我们的$status$里面，1的数量，实际上已经反应了我们安排人员的数量i了，因此，我们只需要**顺序**枚举$status$的**前置状态**pre即可，因为在上文所述的式子$status \oplus (1<<j)$中，如果j是i**能够选择的**，那么一定有$status \oplus (1<<j) < status$，这样就可以省去枚举人数$i$的一维循环，让我们的时间复杂度降为$O(n2^n)$

T2 最小中位数 median (ABC203D)

方法一：暴力枚举

暴力枚举右下端点，将 $k \times k$ 矩阵中的元素取出，然后排序。时间复杂度为 $O(n^2k^2\log k)$，超时。

方法二：二分答案+二维前缀和

我们不妨转变思路，从枚举矩阵到二分中位数。判断一个值能否成为一个 $k \times k$ 矩阵的中位数。

如何判断是否为中位数?

假设一个数 x ，如果 $a_{i,j} > x$，那么另一个二维矩阵中的 $b_{i,j}$ 便设为 1，否则设为 0。这样便将矩阵 a 转变成了一个零一矩阵 b。对其求前缀和，这样就可以求得 b 中任意一个 $k \times k$ 矩阵中 1 的个数。如果其中 1 的个数小于等于 $\left\lfloor \frac{k^2}{2} \right\rfloor$ 那么将上界调为 x，否则将下界调整为 $x+1$。时间复杂度为 $O(n^2 \log n)$。

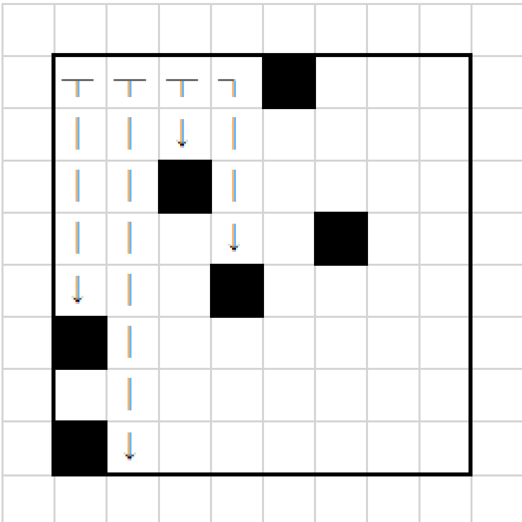
T3 网格上的车 rook (ABC186F)

只有两种方法可以在两次移动或更少的情况下到达一个方格:

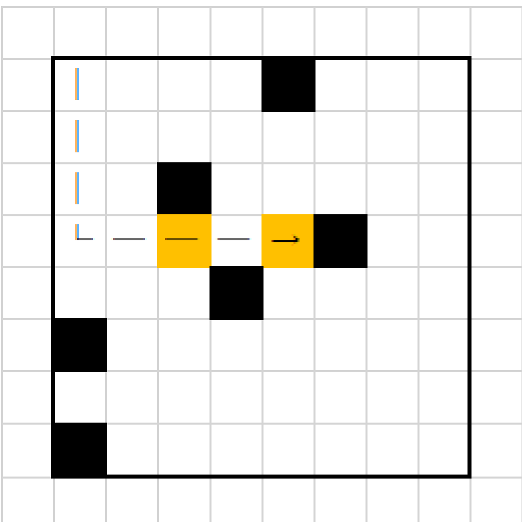
- 向下移动，然后向右移动
- 向右移动，然后向下移动

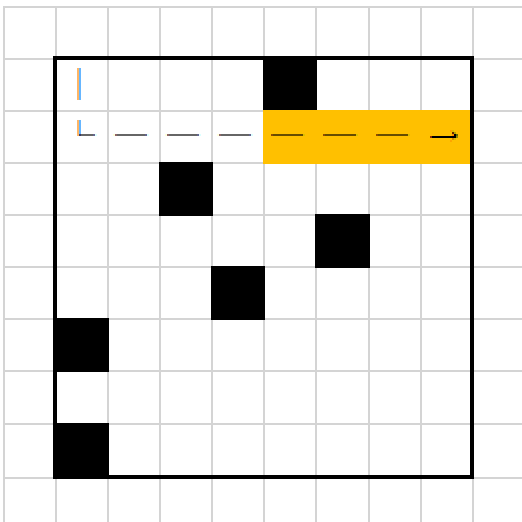
（在上述每次移动中，允许移动零个方格）。由于我们可以通过两种方法到达某些方格，我们必须小心不要重复计算这些方格。

我们可以在 $O(M+W)$ 的总时间内找到后一种方法可以到达的方格，方法是找到每一列中最上面的障碍物。



前一种方法可以用类似的方式找到；但这一次，我们只需要计算那些无法通过后一种方法到达的方格。当且仅当上方有障碍物时，一个方格才能仅通过后一种方法到达（我们认为第一行最左边障碍物右侧的每个方格都有障碍物）。





因此，对于每一行，我们可以用线段树或树状数组来管理每一列是否有障碍物，这样我们就可以在 $O((H+W)\log W)$ 的总时间内计算出答案。

T4 抗火药水 potion (CF543D)

题意

给定一颗 n 个点的无根树，当把一个点 x 设置为起点（树根）时，要求从 x 到任意别的点的路径上有不超过 1 条未铺装的边，现在问你当每一个点作为起点的铺装方案数。

思路

考虑设根为 x，那它就是一个简单的树形 dp，令 dp_u 表示以 u 为根的子树内满足条件的铺装方案，可以发现如果 $u\to v$ 的这条边不铺装的话，那么以 v 为根的子树内的所有边都必须铺装，并且所有 u 的儿子 v 是互不影响的。所以得 dp 方程：

$$dp_u=\prod_{v\in son_u}(dp_v+1)$$

（铺装 $u\to v$ 边，方案数 $1\times dp_v$；不铺装 $u\to v$ 边，方案数 1×1）

由于 $n\le 2\times 10^5$，所以我们显然不能枚举每个点作为根。

这时候就要换根 dp，考虑当根节点从 u 变为儿子 v 时：

- 对于 u 节点，缺少了子树 v，所以 $dp_u=\prod_{k\in son_u,k\neq v}(dp_k+1)$。注意不能用逆元（测试点 $5,6$），只能用前后缀积，因为 0 没有逆元。
- 对于 v 节点，多了它父亲 u 这颗子树，所以 $dp_v=dp_v*(dp_u+1)$，这里的 dp_u 当然是更新后的值。

（如果有同学被第 19 个点（菊花图）或第 20 个点（两条链）卡住了，那么非常抱歉）

时间复杂度：$O(n)$。

T5 城市路径 city（COCI 2008-2009 #3 T6 NAJKRACI）

此题是一个比较套路的题目，若要解决此问题，需要想到以下两个条件：

- 对于任意一条最短路$u->v$，那么，这条路径上的任意一个子路径$u'->v'$也都是一条最短路 (若$u'->v'$不是最短路，那么，一定能找到另一条路径$u_2->v_2$使得其长度小于$u'->v'$，那么，使用这条路径一定会使得答案$u->v$更短，那么原来得到的路径就不是最短路。因此，题目矛盾，证明原结论成立)
- 因为所有城市之间的边权值都为正，因此，在最短路中一定不会存在环。

根据这两个条件，我们可以得出结论：若一条边在最短路图上，则一定有$dis[u]+d_{u,v}=dis[v]$，这样，就可以删除掉那些对答案没有贡献的边。删掉这些边之后，就得到了一个DAG图。

在得到了DAG图之后，因为没有环，故可以用拓扑排序，找到从源点到任意一个点i的最短路径数量$cnt1[i]$，并在反图中跑一个从终点到任意一个点i的最短路径数量$cnt2[i]$。对于任意一条边$d_{u,v}$，答案就为$cnt1[u]\times cnt2[v]$

T6 括号 bracket

算法 1 $\sum|s|,\sum n\le 10$

数据范围很小，考虑先枚举在队首插入和在队尾插入的次数，然后用 $O(2^n)$ 的暴力分别求解。每次 dfs 结束后暴力判读长度为 $|s|+n$ 的字符串，看前缀左括号数量是否均不小于右括号数量。

单次时间复杂度 $O(n\cdot|s|\cdot 2^{n+1})$，期望得分 $8\ pts$。

算法 2 $\sum n\le 10$

此时 $|s|$ 的范围很大，每次 $O(|s|)$ 的字符串判断会超时。可以发现，我们只需要统计给定初始字符串中未被匹配的左括号和右括号数量，就可以跳过每次 $O(|s|)$ 的枚举。

单次时间复杂度 $O(2^{n+1}n^2)$，期望得分 $16\ pts$。

算法 3 $\sum|s|,\sum n\le 1000$

是单次 $O(n^2)$ 的 dp 做法，与正解关系不大，这里略过。

算法 4 特殊性质 B

注意到 s 已经是合法的了。就只需要求长度为 n 的合法括号序列数量，想到卡特兰数，然后枚举队首和队尾的断点，一共是 $n+1$ 种方案。答案就是：

$$(n+1)\times\left[\binom{n}{\frac{n}{2}}-\binom{n}{\frac{n}{2}-1}\right]$$

整体时间复杂度 $O(n)$，来自于预处理，期望得分 $16\ pts$。

算法 5 特殊性质 B

考虑转化问题：

> 初始时在 $(0,0)$，每次可以移动到 $(x+1,y+1)$ 或 $(x+1,y-1)$。求在不碰到 $y=-1$ 这条直线的情况下，询问走到 $(n,0)$ 的方案数。

上述问题将左括号转化成了 $(x+1,y+1)$，将右括号转化成了 $(x+1,y-1)$。这是经典的格路计数问题。

方案数就是用总方案数减去经过直线 $y=-1$ 的方案数。这两个都是好求的。

算法 6 正解

根据算法 2 的优化，我们可以先遍历一遍 s，求出其到达的最低纵坐标 $-p$ 和终点纵坐标 $-k$。

我们就相当于在算法 5 转化的问题上多添加了一个传送操作：

- **必须进行仅一次**传送：若当前纵坐标 $y\ge p$，则可以移动到 $(x,y-k)$。

我们再转化将终点传送上来，即变为从 $(0,0)$ 走到 (n,k)。

我们认为一个点可以传送，当且仅当点 (x,y) 满足：

- (x,y) 在路径上，且 $y\ge p$。
- 之后在 (x,y) 后的点均在直线 $y=k-1$ 之上。

只要求所有路径的可传送点（即向队首和队尾添加括号的分界点）数量和即可。可以做到 $O(n^2)$。

让我们将可传送点分成两类来讨论：

- 可传送点 (x,y) 满足 $y=p$。
- 可传送点 (x,y) 满足 $y>p$。

我们先依次枚举每一个第一类可传送点，有 $\sum\limits_{i=0}^{n}$[从 $(0,0)$ 到 (i,p) 且不碰到直线 $y=-1$ 的方案数] $\times$ [从 (i,p) 到 (n,k) 且不碰到直线 $y=k-1$ 的方案数]。

我们将一段连续的第二类点分成一组（也就是 y 一直大于 p）。

我们可以枚举每一组的左端点 (j,p) 和右端点 (i,p)，可得答案为

$\sum\limits_{i=0}^{n}\sum\limits_{j=0}^{i-1}$[从 $(0,0)$ 到 (j,p) 且不碰到直线 $y=-1$ 的方案数] $\times$ [从 $(j+1,p+1)$ 到 $(i-1,p+1)$ 且不碰到直线 $y=p-1$ 的方案数] $\times$ [从 (i,k) 到 (n,k) 且不碰到直线 $y=p-1$ 的方案数] $\times (j-i-1)$。

让我们将所有含 j 的项单独讨论，考虑它们的组合意义。

我们可以将这三项看成从 (i,p) 出发到 $(0,0)$，不碰到直线 $y=p+1$ 第一次与直线 $y=p$ 相交的位置为 j，则代价为 $(i-j-1)$。

引理

可以发现，只要我们将上面的 (i,p) 看作原点就可以得到以下问题：

- 从 $(0,0)$ 到 (a,b)，过程中不能碰到直线 $y=-1$。
- 设除 $(0,0)$ 外，第一次与 $y=0$ 相交的点 $(i,0)$，则权值为 $i-1$。

则所有合法路径的权值和为 [从 $(0,0)$ 到 (a,b) 与直线 $y=-1$ 相交的方案数]。

证明

假设一条路径在 $(i,0)$ 时第一次与 $y=0$ 相交。

则答案为 $\sum\limits_{i=1}^{a-1}$[从 $(1,1)$ 走到 $(i-1,1)$ 且不碰到直线 $y=0$ 的方案数] $\times (i-1) \times$ [从 $(i,0)$ 到 (a,b) 且不碰到直线 $y=-1$ 的方案数]。

考虑到从 $(0,0)$ 到 (a,b) 且与直线 $y=-1$ 相交的方案数为：

$\sum\limits_{i=1}^{a-1}$[从 $(0,0)$ 走到 $(i-1,-1)$ 的方案数] $\times$ [从 $(i,0)$ 到 (a,b) 且不碰到直线 $y=-1$ 的方案数]。

只需要证明 [从 $(1,1)$ 走到 $(i-1,1)$ 且不碰到直线 $y=0$ 的方案数] $\times (i-1)=$ [从 $(0,0)$ 走到 $(i-1,-1)$ 的方案数]。

可以知道，从 (a,b) 到 (c,d) 的方案数为 $\binom{c-a}{\frac{c-a+d-b}{2}}$。

则从 (a,b) 到 (c,d) 且碰到直线 $y=k$ 的方案数为 [从 (a,b) 到 (c,d) 的方案数] $-$ [从 (a,b) 到 $(c,2k-d)$ 的方案数]。

将组合数代入即可得证。

所以第二类传送点的答案就为：

$\sum\limits_{i=0}^{n}$[从 $(0,0)$ 到 (i,p) 且碰到直线 $y=-1$ 的方案数] $\times$ [从 (i,k) 到 (n,k) 且不碰到直线 $y=p-1$ 的方案数]。

则总的答案为：

$\sum\limits_{i=0}^{n}$([从 $(0,0)$ 到 (i,p) 且碰到直线 $y=-1$ 的方案数] $+$ [从 $(0,0)$ 到 (i,p) 且不碰到直线 $y=-1$ 的方案数]) $\times$ [从 (i,k) 到 (n,k) 且不碰到直线 $y=p-1$ 的方案数]。

然后有 $ans=\sum\limits_{i=0}^{n}$ [从 $(0,0)$ 到 (i,p) 的方案数] $\times$ [从 (i,p) 到 (n,k) 且不碰到直线 $y=p-1$ 的方案数]。

我们将后一项拆开得到：

$ans=\sum\limits_{i=0}^{n}$ [从 $(0,0)$ 到 (i,p) 的方案数] $\times$ ([从 (i,p) 到 (n,k) 的方案数] $-$ [从 (i,p) 到 $(n,k-2)$ 的方案数])。

可以发现，将括号拆开后，前后两项的形式相同，均为合法路径与直线 $y=p$ 的相交次数。

可以发现，上述问题在给定终点 (a,b) 时有 $\sum\limits_{i=\frac{a-b}{2}+p+1}^{a+1}\binom{a+1}{i}$。

后缀和相减可以得到答案就是 $ans=\binom{n+1}{\frac{n-k}{2}+p+1}$。

可以在整体 $O(n+\sum|s|)$ 的复杂度内解决问题，期望得分 $100\ pts$。