

2022“杭电杯”中国大学生算法设计超级联赛 (8) -题解

by willow

感谢大家参加了这次比赛!

同时也为一些考虑不周的地方道歉, orz。

希望多多向我们提出意见!

最后感谢艾泽拉斯天团的友情演出!

1001

题意:

一个 01 序列, 可以任意翻转奇数长度的区间, 求能达到的最小字典序。

题解:

只使用长度为 3 的翻转, 那么位置奇偶性相同的位置可以随便换。对奇数位置和偶数位置, 分别把 0 放到前面、1 放到后面即可。

时间复杂度 $O(n)$ 。

1002

题意:

一个数列, 保证所有数互不相同。现在要把数列划分成若干段, 每一段视为 1-base 的数组, 要求段内最大值在奇数下标, 段内最小值在偶数下标。求划分的方案数。

题解:

设 $dp[i]$ 表示前 i 个位置的划分方案数, 如果区间 $[j, i]$ 是 ridiculous 的, 则 $dp[i]$ 可以从 $dp[j - 1]$ 转移。

从小到大枚举 i , 用两个单调栈分别维护以 i 为结尾的区间的最大值、最小值的位置。设维护最大值的单调栈中相邻两个位置是 x, y , y 为奇数 / 偶数, 则区间 $[x + 1, y]$ 中的下标为奇数 / 偶数的所有位置 j 满足区间 $[j, i]$ 中的最大值在奇数位置, 此时相应的 dp 值可以用于转移。对于维护最小值的单调栈同理。

在维护单调栈的同时维护每个位置是否满足最大值、最小值的转移条件, 转移就是取所有满足了 2 个转移条件的位置。用两个线段树分别对奇数和偶数位置维护满足的转移条件的数量, 线段树中每个区间要维护区间内满足的最多的转移条件的数量, 以及满足最多的转移条件的所有位置的 dp 值之和。

时间复杂度 $O(n \log n)$ 。

1003

题意：

给定一个字符矩阵，求用只能往右往下走的回文路径覆盖的方案。

题解：

因为每次覆盖需要用回文串来覆盖，且矩阵大小只有 $6 \cdot 6$ ，因此可行的状态实际上不是很多，可以直接预处理出所有的可行转移，然后直接记忆化搜索即可（ $6 \cdot 6$ 的全 a 矩阵可行的状态数只有大约 $2 \cdot 10^6$ 种）。

注意答案可能爆int，需要用long long存。

1004

题意：

二维平面上， $[0, n] * [0, n]$ 的整点中去掉 $(0, 0)$ 之后最少需要多少条直线使得每个点至少有一条边覆盖。

题解：

答案为 $2 * n$ 。

证明：我们考虑如果不切 $x = n$ 和 $y = n$ ，那么同一条直线最多穿越两个边界上的点，总共有 $4n$ 个边界上的点，所以至少要 $2n$ 条直线。

我们考虑实际操作时切了一些边界直线，比如说我们切了

$x = n, x = n - 1, \dots, x = n - k_1, y = n, y = n - 1, \dots, y = n - k_2$ 我们没有切 $x = n - k_1 - 1$ 和 $y = n - k_2 - 1$

那么我们可以认为我们先花了 $k_1 + k_2 + 2$ 条直线将范围限定到了 $[0, n - k_1 - 1] \times [0, n - k_2 - 1]$ 的范围，且不切 $x = n - k_1 - 1$ 和 $y = n - k_2 - 1$ 。剩下的每条直线最多穿越两个边界上的点，所以至少需要 $n - k_1 - 1 + n - k_2 + 1$ 条直线才能完全覆盖。

所以总共至少需要 $2n$ 条直线，且 $x = 1, x = 2 \dots x = n$ 和 $y = 1, y = 2 \dots y = n$ 这 $2n$ 条直线是一组可行解。所以答案为 $2n$ 。

1005

题意：

一条链，每个点上有一个数 a_i ，每条边上有一个质数 b_i 。一开始在某个点上，有一个空背包，走到一个点上可以把它的质因子放进背包，一条边如果背包里有那个质数就可以走。多组询问求从 x 出发能否走到 y （即求每个点能走到的最大范围）。

题解：

求出从每个点出发能到达的最大的范围，即可 $O(1)$ 回答所有询问。

先从大到小枚举，预处理每个点只向右走的最大范围，如果能向右走一步就先取 $i+1$ 的范围，然后沿着之前预处理的范围向后跳。向后跳的次数是均摊 $O(1)$ 的。

然后从小到大枚举，求每个点能走到的最大范围。

如果 i 能走到 $i + 1$ ：

如果 $i + 1$ 已经预处理的向右的范围无法让它走回 i 则取预处理的答案;

否则 $i + 1$ 和 i 范围相同。

如果 i 不能走到 $i+1$:

先取预处理的范围, 然后向左向右扩展, 都沿着已知的范围暴力跳即可。向左向右跳的次数都是均摊 $O(1)$ 的。

判断一个范围内是否存在某个质因子, 方法有很多, 标程采用的是对每个质因子维护一个 vector 存它出现的所有位置, 在 vector 上二分。

时间复杂度 $O(n \log n)$ 。

1006

题意:

模拟飞行棋, 输出谁会赢且按分数从大到小输出每个人的分数信息。

题解:

由于它跟现实的飞行棋的规则会有一些差别, 所以需要自己阅读题面的规则。

需要注意的可能有以下几点:

1. 在起飞区棋子不能直接叠起来一起走, 所以会对某些人的策略有影响。
2. 游戏结束的时候注意每个人最后骰子点数的连续段有没有被算入得分。
3. 棋子在属于自己颜色的Final stretch上的最外围是会被保护的, 即使在地图上看起来如果两个不同颜色的棋子落在了那里的同一个点, 如果其中一方被保护了那么是会产生战斗的。
4. 连续10次无法操作后会获得一次launch的机会, 但不会对点数有任何的操作。
5. 注意在Final stretch里起点和终点的距离可能是负数, 因此在Carol的策略中要注意最大值的记录初始不能设为0。

1007

题意

给出一个排列 p , 把每个位置视为点, 建一个无向图, i, j 之间的边权为 $|i - j| * |p_i - p_j|$ 。求这个图的最小生成树。

题解:

依次连接 $i, i + 1$ ($1 \leq i < n$), 这样的生成树每条边边权都 $\leq n - 1$, 因此存在一种最小生成树中也只有边权 $\leq n - 1$ 的边 (Kruskal 算法的原理)。

$|i - j| * |p_i - p_j| \leq n - 1$ 意味着 $|i - j|$ 和 $|p_i - p_j|$ 必有至少一个 $\leq \sqrt{n - 1}$, 因此可以在 $O(n\sqrt{n})$ 的时间复杂度内找到所有这样的边, 然后使用 Kruskal 算法求出最小生成树即可。

总时间复杂度 $o(n\sqrt{n}\alpha(n))$ 。

1008

题意：

求一棵树的最大分离集的大小

题解：

令 $dp[x][0]$, $dp[x][1]$, $dp[x][2]$ 分别表示仅考虑 x 这个子树, x 号点未选, x 号点选择了但是度数为 0, x 号点选了且度数为 1 的选且 x 子树内部满足分离集定义选择点数的最大值。

直接 dp 即可!

详见 std 代码。

1009

题意：

一棵树, 有 m 次 access 操作依次执行, 每次 access 操作会把某个点到根的路径上所有边染成某个颜色, 并且把这条路径上连出去的其它边的颜色清零 (其实差不多就是 LCT 的 access)。现在每个操作有一定的概率成功, 一定的概率失败, 求最终所有边上的颜色之和的期望。

题解：

对每个点 u , 它指向所有孩子的边最终最多只有一条是有颜色的, 条件是在 u 的子树中某个点 (不包括 u) 上的一次 access 操作成功, 且比这次 access 晚的其它 access (这里要包括 u 上的 access) 都失败。

使用线段树合并, 以执行的顺序为下标, 维护 u 子树内的所有操作。线段树上每个区间结点维护两个值: prod 表示区间内所有 access 的 $(1 - p_i)$ 的乘积, sum 表示区间内每个 access 的 $c_i * p_i$ 与其右边的所有 access 的 $(1 - p_i)$ 的乘积。

u 的子树内都合并完之后, 答案先加上线段树根的 sum, 此时的 sum 混入了 u 上的所有操作, 需要额外减去, 在线段树上 DFS 可以 $O(\log n)$ 求出单个 access 此时对 sum 的贡献。

时空复杂度均为 $O(n \log n)$ 。

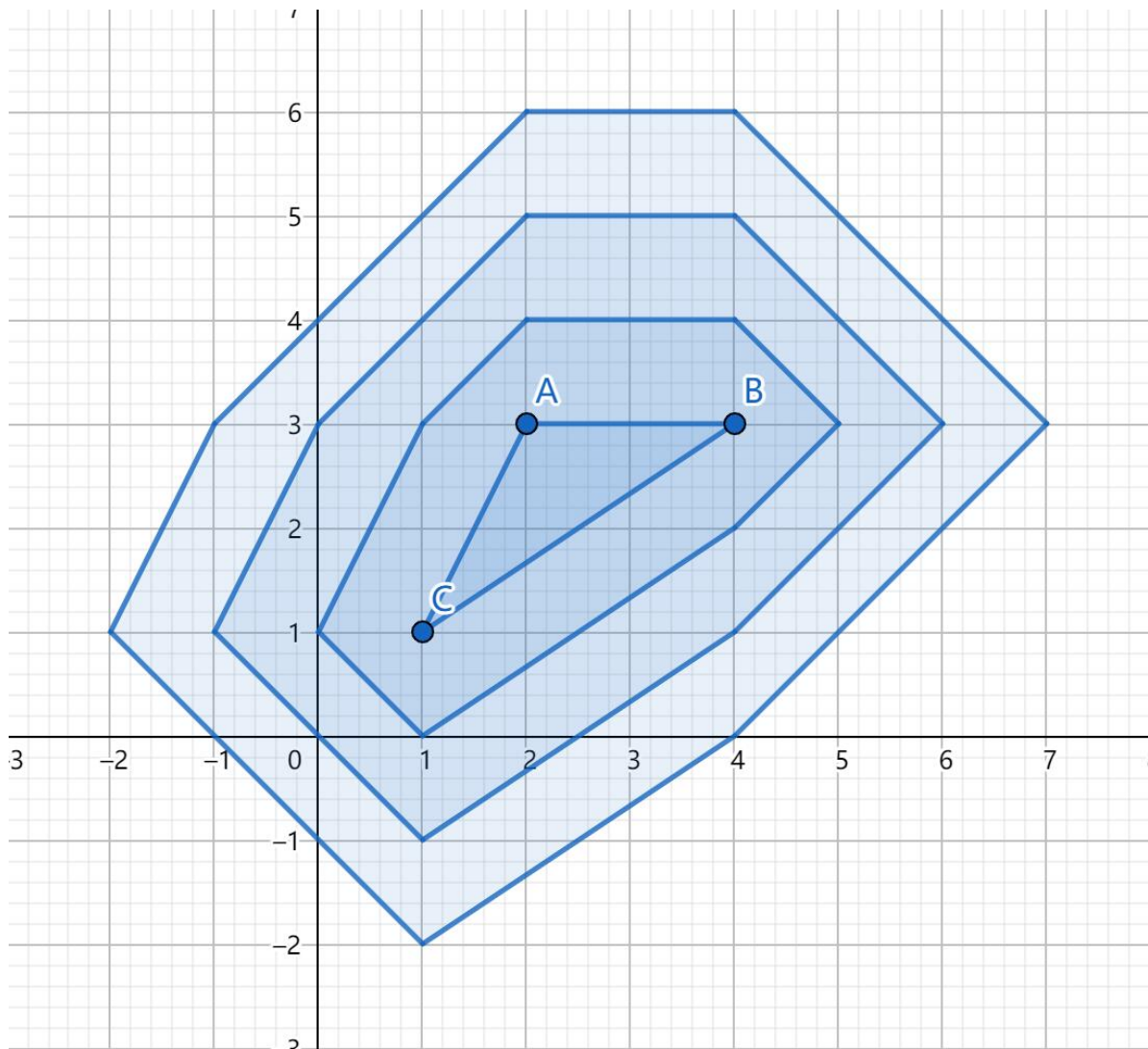
1010

题意：

平面上原来有 n 个点, 每一秒每个存在的点会向自己相邻的四个方向扩展一个点 (如果已经存在则不会生成), 多组询问 t , 求 t 秒后所有点形成的凸包面积。

题解：

我们考虑每次扩展完后凸包会产生什么变化, 样例 1 的情况如下图所示:



可以发现在每一秒扩展的时候，原凸包上与坐标轴不成 45° 角的所有边的长度是不会有改变的，而成 45° 角的所有边（只有4条）长度每一秒会+1（如果一开始的凸包上没有，则会在第一秒加入一条）。

再考虑在这样的变化下，凸包的面积会有什么变化：

1. 对于那些与坐标轴不成 45° 的所有边，它们每秒会往外平移1的距离，而方向是垂直与坐标差较大的那一维，因此每一秒多出来的面积为 $\max(|s.x - t.x|, |s.y - t.y|)$ ，其中 s, t 为该凸包线段起始点与终点。
2. 对于那些与坐标轴成 45° 的所有边，它们每秒会往外平移并且长度+1，假设一开始边的长度为 $\sqrt{2}k$ （即两维坐标差都是 k ），且经过了 t 秒，那么长度会变成 $\sqrt{2}(t + k)$ ，而增加的面积则是 $\frac{1}{2}((t + k))^2 - k^2 = \frac{1}{2}t^2 + kt$ 。

因此对于每一次询问的 t ，我们只需要 $O(1)$ 求出答案即可。

1011

题意：

一个 $n * m$ 的长方形，可以沿水平或竖直方向画若干条线，每条线的两端点都在长方形的边界上。要求这些线划分出的每个小长方形面积都 $\geq k$ ，求最多可以画几条线。

题解：

枚举小长方形的某个边长的最小值，由此可以求出另一个边长允许的最小值，然后求出两个方向分别最多能画几条线。

时间复杂度 $O(k)$ 。

1012

题意：

构造一棵有根二叉树，使得每个点要么是叶子要么有两个儿子，且深度之和为 k ，求先优先最小化点数深度数量序列字典序最小的树的深度数量序列。

题解：首先，考虑 k 为奇数和 $k=4,8,14$ 的情况是无解的，其他情况下一定有解。

其次，我们可以构造一条尽量长的链，剩如果还有剩余的度数说明至少还需要2个点。这样我们能够算出最少点数。

最后，我们贪心构造答案。我们定义树的第 i 层为深度为 i 的所有点。

那么叶子要么有两个儿子，且深度之和为 k 的性质可以等价于每一层的点数为偶数（除了第0层），并且每一层的点数小于等于上一层点数的两倍。

由于点数是 $O(\sqrt{n})$ 级别的，我们考虑字典序的贪心。

我们每次确定某一个深度的点数，我们从小到大枚举。

判断的方法是，剩下的所有点放在尽量小的深度，所有点的深度之和是否超过了剩余的深度之和。

正确性证明比较简单，大家可以自行思考一下。

大致思路：由于我们求的是字典序最小值，通过这一点证明我们无需判断所有点的深度之和是否小于等于剩余的深度之和。然后证明在有解的情况下，假设深度之和最小为 L ，最大为 R ，则 L 到 R 之间相差二的所有解都可以达到。

复杂度 $O(\sqrt{n} \log n)$ 。因为假设我们现在确定的是深度为 i 的点，那么从深度为 $i+1$ 的点开始，每一层最大能放点的数目是上一层的两倍，所以我们最多枚举 $O(\log(\sqrt{n}))$ 层就会结束。

1013

题目：

一个长 m 的数组，每个位置的数 $\in [1, n]$ ，求不存在任何一个长 n 的子区间是 $1 \sim n$ 的排列的方案数。

题解：

反过来求有 $1 \sim n$ 的排列的方案数。枚举从左到右第一次出现排列的位置，容斥减去更前出现排列的方案数。

设 f_i 表示第一个排列出现在以 i 开头的子区间时 $[1, i+n-1]$ 位置的方案数，三种转移：

1) $f_i = n^{i-1} * n!$ (随便填)

2) $f_{i-} = f_j * n^{i-j-n} * n!$, $j+n-1 < i$ (更前的排列不包含 i)，前缀和优化

3) $f_{i-} = f_j * (i-j)!$, $j+n-1 \geq i$ (更前的排列包含了 i)，分治 NTT 优化

答案为 $ans = n^m - \sum_{i=1}^{m-n+1} f_i * n^{m-i-n+1}$ 。

时间复杂度 $O(n \log^2 n)$ 。

还能再优化：

令 $F(x) = \sum_{i=0}^{\infty} f_{i+1}x^i$, $G(x) = \sum_{i=0}^{\infty} n^i n! x^i$, $H(x) = \sum_{i=1}^{n-1} i! x^i$, 则上述转移可以写成生成函数形式:

$$F(x) = G(x) - F(x)G(x) \cdot x^n - F(x)H(x)$$
$$F(x) = \frac{G(x)}{1 + x^n G(x) + H(x)}$$

多项式求逆, 时间复杂度 $O(n \log n)$ 。

两种做法实际运行时间相差不大, 都可以通过本题。