

试题泛做报告

厦门双十中学 汪文潇

1. codeforces 306C White, Black and White Again

题目大意:

在 n 天内有 w 件好事和 b 件坏事, 要求前若干天发生好事, 再若干天发生坏事, 最后若干天发生好事, 每天都有事发生, 求方案数模 10^9+9 。

算法讨论:

枚举有多少天发生坏事, 如果有 i 天发生坏事, 该情况的方案数就是

$$C(n-i-1, w-1) * C(i-1, b-1) * w! * b! * (n-i-1)$$

对每种情况求和即可。

时间复杂度: $O(n+b+w)$

空间复杂度: $O(n+b+w)$

2. codeforces 261E Maxim and Calculator

题目大意:

有 2 个变量 a, b , 初值 $a=1, b=0$, 每次可以让 $a=a*b$ 或 $b=b+1$, 问能在 p 步内让 a 变成的值在 $[1, r]$ 内有多少个。

算法讨论:

对于一个数 x , 把 x 分解成若干数的乘积, 就对应了一种方案, 代价为其中最大的数+数的个数。注意到 $p \leq 100$, 所以 x 如果能被表示它的所有质因子一定都小于 p , $p=100$ 时, 在 $[1, 10^9]$ 内只有不到 $3 \cdot 10^6$ 个质因子小于 p 的数, 所以我们可以把这些数都生成出来, 然后从小到大枚举式子中最大的数, 然后暴力 dp 求出当前每个数最少要由几个不大于当前枚举数的数相乘才能得到。

时间复杂度: $O(p \cdot f(r, p))$, $f(r, p)$ 表示在 $[1, r]$ 内最大质因子小于 p 的数的个数。

空间复杂度: $O(f(r, p))$

3. Codeforces 283E Cow Tennis Tournament

题目大意:

有 n 头牛, 每头牛有能力值, 能力值互不相同, 一开始能力高的会战胜能力低的, 然后进行 k 次修改每次使得能力值在 $[a_i, b_i]$ 之间的奶牛的比赛结果逆转, 问最后有多少无序三元组 (a, b, c) 满足 a 胜 b , b 胜 c , c 胜 a 。

算法讨论:

总共的三元组个数-满足其中一个胜了另 2 个的三元组个数就是所求的答案了, 那么我们只要计算出每只奶牛的胜利数 $win[i]$, 答案就是 $C(n, 3) - \sum C(win[i], 2)$, 每次的修改实际上是对结果矩阵的一个子矩阵进行翻转, 然后我们要统计矩阵的每一行的和, 可以用扫描线+线段树解决。

时间复杂度: $O(n \log n + k \log k + k \log n)$

空间复杂度: $O(n+k)$

4. Codeforces 319D Have You Ever Heard About the Word?

题目大意:

给定一个长度为 n 的字符串，不断每次把其中最短的“重复块”中最左侧的删除，直到无法删除为止，求最后的字符串。“重复块”指的是由一个字符串与自身连接而成的字符串，例如: `abcabc`，而 `abcabcabc` 就不是。

算法讨论:

注意到在过程中被删除的块长度一定是单调不降的，我们从小到大枚举被删除的块的长度，对于给定的长度 l ，我们可以用 $n/l * \log n$ ($\log n$ 是二分+hash 的判断复杂度) 的复杂度判断是否至少存在一个重复块，如果存在，我们就用 $O(n)$ 的时间暴力把该长度的重复块都删掉。注意到不同的至少可以删一次的长度最多只有 $O(n^{\theta.5})$ 种，因而复杂度是 $O(n \log^2 n + n^{1.5})$ 的。

时间复杂度: $O(n \log^2 n + n^{1.5})$

空间复杂度: $O(n)$

5. Codeforces 317C Balance

题目大意:

给定一个 n 个点， e 条边的无向图，每个点有初始水量 $a[i]$ ，最终水量 $b[i]$ 和容量限制 $v[i]$ ，每次操作可以从一个点通过一条边将任意水量转移到另一个点，但要保证容量限制。要做的事给出一种操作数 $\leq 2 * n^2$ 的方案或输出无解。

算法讨论:

首先对于每个连通块要求 $\sum a[i] = \sum b[i]$ ，否则显然无解。然后我们对于每个连通块取一棵生成树，然后对上面当前 $a[i] < b[i]$ 的点分别进行调配，我们把所取的点当成树根，然后对于它的每个孩子的子树，如果子树内的 $\sum a[i] > \sum b[i]$ ，我们就要从这个子树内抽取水量到当前的点，然后递归到子树内分配水量。

时间复杂度: $O(n^3)$

空间复杂度: $O(n+e)$

6. Codeforces 277D Google Code Jam

题目大意:

有一场比赛有 n 道题，每题有 `small` 和 `large` 两个部分，每部分有对应得分，要先完成 `small` 才能做 `large`，`small` 保证一次通过，`large` 部分有 p 的概率 `fail system test`。你的得分为所有部分的得分和，你的罚时为最后一次提交正确解的时间。让你安排解题顺序，使得期望得分最大的情况下，期望罚时最小。

算法讨论:

首先解题顺序是不影响期望得分的，只影响期望罚时，我们考虑我们的解题顺序中相邻的某 2 道题 a, b ，如果交换他们的顺序不能使期望罚时更优，那一定有 $a.t2 * (1 - a.f) * b.f + (a.t2 + b.t2) * (1 - b.f) \leq$

$$b.t2 * (1 - b.f) * a.f + (a.t2 + b.t2) * (1 - a.f)$$

(其中 f 表示 `large` 不通过的概率， $t2$ 表示 `large` 在 `small` 的基础上需要的额外时间)

那我们就以这个为比较函数把题目排序，然后 dp。

时间复杂度: $O(n \log n + nt)$

空间复杂度: $O(n+t)$

7. Codeforces 263E Rhombus

题目大意:

给定 $n \times m$ 的矩阵和 k , 找到满足下面条件的 (a, b)

1. $k \leq a \leq n - k + 1$

2. $k \leq b \leq m - k + 1$

3. $k \leq x \leq n - k + 1$ 且 $k \leq y \leq m - k + 1$ 的范围内定义函数 $f(x, y) = \sum \sum a[i][j] * \max(0, k - \text{abs}(i - x) - \text{abs}(j - y))$ 的最大值是 $mval$ 。对于我们要找的整数对必须满足下列等式 $f(a, b) = mval$ 。

算法讨论:

这是我看到的一种精美的解法。

我们定义 $s[i][j]$ 表示 (i, j) 左上角的子矩阵内数值的和,

$sl[i][j] = \sum s[i-t][j-t] \quad (t < \min(i, j) \& \& 0 \leq t < k)$

$sr[i][j] = \sum s[i-t][j+t] \quad (t < \min(i, m-j) \& \& 0 \leq t < k)$

这 3 个都可以在 $O(n^2)$ 的时间内递推求得。

我们枚举 x 和 y , 而此时

$f(x, y) = sr[x+k-1][y] + sr[x-1][y-k] - sl[x-1][y+k-1] - sl[x+k-1][y-1]$, 我们可以 $O(1)$ 计算出了。

时间复杂度: $O(nm)$

空间复杂度: $O(nm)$

8. Codeforces 301C Yaroslav and Algorithm

题目大意:

要求你按照题目给定的规则构造一个算法, 使得这个算法对于给定的输入字符串, 能够输出这个字符串对应的数字+1 的字符串。

算法讨论:

注意到我们能用的字符除了 0~9 还有? , 那么我们大可用? 做一个指针来模拟高精数+1 的过程。

9. Codeforces 235E Number Challenge

题目大意:

给定 a, b, c , 定义 $d(n)$ 表示 n 的约数个数, 计算

$\sum \sum \sum d(i * j * k) \quad (1 \leq i \leq a, 1 \leq j \leq b, 1 \leq k \leq c)$ 模 2^{30} 的值

算法讨论:

出题人的做法是 dp, 利用了有效状态很少的性质, 而当时有人给出了这样的一个结论: (证明略)

$ans = \sum \sum \sum (a/i) * (b/j) * (c/k)$

$(1 \leq i \leq a, 1 \leq j \leq b, 1 \leq k \leq c, \text{gcd}(i, j) = \text{gcd}(i, k) = \text{gcd}(j, k) = 1)$

那么我们可以预处理 $f[i] (1 \leq i \leq c)$ 表示 $\sum c/j (j \text{ 是 } i \text{ 的倍数})$, 然后用 $f[i]$ 通过莫比乌斯反演计算出 $w[i] (1 \leq i \leq a * b)$ 表示 $\sum c/j (\text{gcd}(i, j) = 1)$ 。

接下来枚举 i, j 并求和即可。

时间复杂度: $O(a * b * \log(a * b) + c \log(c))$

空间复杂度: $O(a*b)$

10. Codeforces 351D Jeff and Removing Periods

题目大意:

对于一个序列, 你的一次操作是这样的:

- 1、从任意位置起到任意位置结束, 每隔 k 个就删除一个, k 可以由你指定, 但要求删除的所有数数值相同
- 2、把未删除的数任意排序

现在给定一个序列, q 次询问, 每次问你该序列的一个连续子序列至少要经过多少次操作才能被全部删除。

算法讨论:

对于一个序列, 我们进行了第一次操作后, 由于我们可以任意排序, 不难发现接下来我们的最少操作次数其实就是剩余的不同数字的种数, 那么我们只需要知道一开始有多少不同数字以及第一步操作后能否删除某一种数字即可。

预处理每个点为起点向右删除所有与它数值相同的数最多能删到的位置以及每个点左边第一个与它相同的数的位置, 然后随意用数据结构维护即可。

时间复杂度: $O(n\log n + q\log n + q\log q)$

空间复杂度: $O(n+q)$

11. Codeforces 241B Friends

题目大意:

给出 n 个整数 $a_1 \sim a_n$, 定义无序二元组 (i, j) 的权值为 $a_i \wedge a_j$ (\wedge 表示亦或), 求前 m 大的二元组权值和。

算法讨论:

对这个 n 个数建 **trie**, 然后在 **trie** 上二分第 m 大的值即可, 时空复杂度都是 $O(n\log^2 A)$ 的。

有一种小优化, 就是我们可以把这 n 个数排序, 然后 **trie** 上的子树就对应序列的一个区间, 然后把这个序列当 **trie** 来跑即可, 空间复杂度降到 $O(n\log A)$, 时间复杂度不变, 但常数较小。

时间复杂度: $O(n\log^2 A)$

空间复杂度: $O(n\log^2 A) / O(n\log A)$

12. Codeforces 323B Tournament-graph

题目大意:

构造一个有 N 个结点的竞赛图, 使得对任意两个结点 u 和 v ($u \neq v$), 从 u 到 v 的最短距离不超过 2。

竞赛图就是基图为无向完全图的有向图 (每对结点之间有一条有向边相连, 且无自环)。

算法讨论:

N 为奇数时, $N=3$ 有解, N 为偶数时, $N=4$ 无解, $N=6$ 的解可以手动构造。

那么我们对于点数为 N 的要求, 根据奇偶先得到一个基本解 ($n=3/n=6$), 然后每次增加 2 个点使得新图仍满足条件即可。

具体方法就是把原来的 n 个点看做一个块, $n+1/n+2$ 各看做一个块, 然后像 $n=3$ 那样将 3 块连成一个环, 就得到了 $n+2$ 的可行解。

时间复杂度: $O(n^2)$

空间复杂度: $O(n^2)$

13. USACO 2008 March Land Acquisition

题目大意:

n 块土地, 每块有长 l 和宽 w , 一次性购买一个集合的土地的价格是这些土地中最大的长*最大的宽, 问把所有土地买下的最小花费。

算法讨论:

把土地按长排序后, 遍历一遍得到一个关于宽的单调队列, 只需要考虑买下这一些就行了。然后就是经典的斜率优化动态规划。

时间复杂度: $O(n \log n)$

空间复杂度: $O(n)$

14. Codeforces 305E Playing with String

题目大意:

双人博弈, 轮流操作, 不能操作的人输。一开始只有一个字符串。

每次可以选择所有字符串中的一个(称为 t)中的某个位置 i , 使得存在一个正整数 $k(0 < i-k, i+k \leq |t|)$ 满足 $t[i-1]=t[i+1]$, $t[i-2]=t[i+2]$... $t[i-k]=t[i+k]$, 然后把这个字符串分成 3 个字符串: $t[1 \sim i-1], t[i \sim i], t[i+1 \sim |t|]$ 。

算法讨论:

预处理初始哪些位置可删, 记录连续可删位置的长度, 然后计算出 1 个连续可删位置的 sg , 然后暴力计算即可。

时间复杂度: $O(n^2)$

空间复杂度: $O(n)$

15. Codeforces 314E Sereja and Squares

题目大意:

在 x 轴上有 n 个点, 第 i 个点坐标 $(i, 0)$, 每个点标上了一个小写或大写英文字母 (没有 'x'/'X')。

这些点是漂亮的, 当且仅当:

- 所有的点可以被分成若干对, 使得每个点恰好属于一一对之中。
- 在每对点中, 横坐标较小的会被标上小写字母, 较大的会被标上对应的大写字母。
- 如果我们在每对点上画一个正方形, 其中已知的一对点会作为正方形的相对的顶点, 它们间的线段会成为正方形的对角线, 那么在所有画出的正方形中不会有相交或触碰的情况。

现在擦掉了一些小写字母和所有大写字母, 问有多少种方法来还原每个点上的字母, 使得还原后这些点是漂亮的。

算法讨论:

实际上问题差不多就是给定若干左括号的括号序列方案数, 然后就 $f[i][j]$ 表示前 i 个中左括号比右括号多 j 个的方案数, 需要常数优化&空间优化。

时间复杂度: $O(n^2)$

空间复杂度: $O(n)$

16. Codeforces 293E Close Vertices

题目大意:

n 个点的树，边有一个非负边权，路径的长度包含的边数，路径的权重是路径包含的边的边权和。

两点“相邻”，当且仅当，存在一条连接该两点的路径，满足该路径的长度小于等于 L ，且权重小于等于 W 。

统计有多少个点对 (u, v) ，满足 $u < v$ ，且 u, v 是相邻的。

算法讨论:

树分治经典题。

时间复杂度: $O(n \log^2 n)$

空间复杂度: $O(n)$

17. Codeforces 238D Tape Programming

题目大意:

给了你一种语言的运行规则（详情还是见题目吧），这种语言需要一个字符串，多组询问，每次询问一开始给定的串 S 的一个连续子串在这个规则下会将每个数字输出多少次。

算法讨论:

我们先模拟整个串 S 的运行情况（如果有一个后缀没有被访问到，那么还需要从这个后缀的开头为起点继续模拟），并记录下 $cnt[i][dir]$ 表示第一次出现当前位置为 i ，下一步的方向为 dir 的时候每种数字输出了多少次。注意到询问的子串对应原串模拟的一段连续时间，那么我们之间头尾相减即可。

时间复杂度: $O(n+q)$

空间复杂度: $O(n+q)$

18. Codeforces 303D Rotatable Number

题目大意:

要找到最大的 $b(1 < b < x)$ ，满足在 b 进制下存在一个长度为 n 的正“可旋转数”（允许有前导零）。

算法讨论:

结论题，这种数可以表示成这样的形式 $(b^{(p-1)}-1)/p$ ，这个有 $p-1$ 位，要求 $p=n+1$ 是素数，且 b 是它的原根，然后就暴力找 $n+1$ 的原根即可。

19. Gcj Final 2009A Year of More Code Jam

题目大意:

这一年有 n 天，有 t 场线上赛，第 i 场线上赛有 m_i 场比赛，每次比赛分别在这场线上赛的第一场比赛开始后的当天、 $d[i][2], d[i][3] \dots d[i][m_i]$ 天，每场线上赛随机等概率的开始于任意一天，求 $\sum f[i]^2 (i=1 \sim n)$ 的期望值（ $f[i]$ 表示这年的第 i 天有多少场比赛）

算法讨论:

暴力维护只考虑前 i 场线上赛的时候的答案和每天的期望比赛数，然后考虑加入第 $i+1$ 场线上赛后对答案的增量，即可。

时间复杂度: $O(m^2 * t^2)$

空间复杂度: $O(mt)$

20. USACO 2009 open Tower of Hay

题目大意:

给定 N 个干草, 高度为 1 , 宽度为 w_i , 可以把若干个放在同一层, 堆成若干层, 满足: 对于任意一层的积木, 他的宽度之和要小于等于他下面那一层的积木 (最底层除外) 并且不允许编号小的放在编号大的的积木上面。求最多能够堆多少层。

算法讨论:

首先有一个结论是对于上面的若干层, 底层最窄和高度最优是可以同时存在的。那么我们就只要 dp 记录最窄的宽带和最高的高度就行了。

时间复杂度: $O(n)$

空间复杂度: $O(n)$

21. Codeforces 321D Ciel and Flipboard

题目大意:

$n*n$ 的矩阵 A , n 是一个奇数, 不妨设 $x=(n+1)/2$ 。每次可以选择一个 x 行 x 列的子矩阵, 并将其中的所有元素乘 -1 。可以操作任意多次。问矩阵数字和的最大值。

算法讨论:

我们设一个 01 矩阵 C 表示 A 的每一个位置最终翻转与否, 那么我们有:

$$C[i][j] \oplus C[x][j] \oplus C[i+x][j] = 0 \quad (i < x, 0 < j \leq n),$$

$$C[i][j] \oplus C[i][x] \oplus C[i][j+x] = 0 \quad (j < x, 0 < i \leq n)$$

证明很容易, 因为每次操作一定修改了式子中偶数个项。

那么我们枚举 $C[1][x] \sim C[x][x]$, 然后就能直接推出第 x 列的 C , 接下来我们对于第 x 行, 贪心的判断每个位置是否翻转即可。

时间复杂度: $O(2^{((n+1)/2)} * n^2)$

空间复杂度: $O(n^2)$

22. USACO 2010 Dec Threatening Letter

题目大意:

给定 2 个字符串 A 、 B , 长度分别为 n 、 m , 要求你把 B 分成尽量少的几段, 使得每段都是 A 的一个子串, 求最少段数。

算法讨论:

贪心, 每次从头开始取尽量长的部分分成一段即可。

实现可以使用后缀数组/后缀自动机。

时间复杂度: $O((n+m) \log(n+m)) / O(n+m)$

空间复杂度: $O(n+m)$

23. USACO 2007 Open Connect

题目大意:

给定 $R*C$ ($1 \leq R \leq 2, 1 \leq C \leq 15000$) 的网格图, 只有相邻的格点可能有边相连, 要求在线支持: 删除一条边、加入一条边、查询 2 点是否连通 (只能经过他们所处的两列之间的边)。

算法讨论:

直接用线段树维护区间端点间的连通性即可, 可以把 R 默认为 2。

时间复杂度: $O(C \log C)$

空间复杂度: $O(C)$

24. USACO 2008 Open Cow Neighborhoods

题目大意:

有 n 只奶牛, 每只奶牛有不同的坐标。两只奶牛属于同一个群, 当且仅当至少满足下列两个条件之一:

1. 两只奶牛的曼哈顿距离不超过 C , 即 $|x_1 - x_2| + |y_1 - y_2| \leq C$ 。

2. 两只奶牛有共同的邻居。即存在一只奶牛 k , 使 k 分别与这两只奶牛同属一个群。

计算有多少个牛群, 以及最大的牛群里有多少只奶牛。

算法讨论:

首先把坐标轴旋转, 另 $x' = x + y, y' = x - y$, 则曼哈顿距离不超过 C 等价于坐标差的最大值不超过 C 。

把奶牛按 x 排序, 然后扫描一遍, 用一个 `set` 维护与当前奶牛 x 的差不超过 C 的集合, 然后加入当前奶牛, 并将当前奶牛与当前集合中按 y 排序时它的前驱和后继进行比较, 如果差不超过 C 则用并查集合并。

最后统计每个并查集即可。

时间复杂度: $O(n \log n)$

空间复杂度: $O(n)$

25. Codeforces 266E More Queries to Array...

题目大意:

你有一个包含 n 个整数 $a[1], a[2], \dots, a[n]$ 的数组。有 2 个操作:

1. 将 $a[1], a[1+1], \dots, a[r]$ 赋值为 x 。

2. 计算 $\sum a[i] * (i - l + 1)^k (l \leq i \leq r)$ 模 $10^9 + 7$, 此处 k 不超过 5。

算法讨论:

用线段树直接维护所求式子展开后的每项即可。

时间复杂度: $O(nk \log n)$

空间复杂度: $O(nk)$

26. Codeforces 338D GCD Table

题目大意:

定义 $G(i, j) = \gcd(i, j)$, 给出一段序列, 是否在 $G[1 \sim n][1 \sim m]$ 的某一行出现过。

算法讨论:

首先我们只要考虑 $x = \text{lcm}(a[1] \sim a[k])$ 作为行号是否可行, 然后这时其实得到了一个关于起点的列号 j 的同余方程组, 解出最小的解判断是否合法即可。

时间复杂度: $O(k \log n + k \log m)$

空间复杂度: $O(k)$

27. Codeforces 325D Reclamation

题目大意:

有一个 $r \times c$ 的地图, 把左边界和右边界粘起来使得形成一个圆柱, 现在要不断地挖去其中的格子, 要求任何时候都存在一条从最上方到最下方的路径(四联通), 如果某次操作不满足要求则不做, 问最后有多少次操作是成功的。

算法讨论:

把地图复制一份接在右侧, 每次加入后如果使得存在 i, j 与 $i, j+c$ 在八联通的情况下连通, 则操作是失败的。并查集维护。

时间复杂度: $O(r \times c \times \log(r \times c))$

空间复杂度: $O(r \times c)$

28. Codeforces 260E Dividing Kingdom

题目大意:

平面上有 n 个点, 让你用 2 条竖直的直线和 2 条水平的直线把平面分成 9 个部分, 每个部分内的点数与 $a[1] \sim a[9]$ 一一对应(顺序可以由你定), 给出方案。

算法讨论:

枚举每个部分分别有多少个点, 然后用可持久化线段树检验是否有解。

时间复杂度: $O(9! \log n + n \log n)$

空间复杂度: $O(n \log n)$

29. Codeforces 339E Three Swaps

题目大意:

给定 n 的一个排列。一次操作是指翻转一个区间。让你给出一个 3 步以内的操作序列来将 $1 \sim n$ 的有从小到大顺序排列变成给定排列。保证有解。

算法讨论:

我们逆向搜索, 对于每次操作 (l, r) , 还原后一定能使 $a[l-1]$ 和 $a[l]$ 相邻或者 $a[r]$ 和 $a[r+1]$ 相邻, 以此剪枝即可。

30. Codeforces 266D BerDonalds

题目大意:

给定一个无向带权联通图, 求图的直径的一半。

算法讨论:

枚举直径的中点所在的边 (u, v) , 设其距离点 u 距离为 x , (u, v) 边权为 len , 那么我们的目标就是最小化 $\max(\min(x + \text{dist}[u][i], len - x + \text{dist}[v][i]))$, 最优决策点一定在该函数的折点, 按 $\text{dist}[u][i]$ 排序, 扫描一遍求出折点即可。注意到我们枚举了 u 以后顺序已经确定, 所以只需要排序 n 次即可。

时间复杂度: $O(n^3)$

空间复杂度: $O(n^2)$

31. USACO 2012 Dec First!

题目大意:

给定 n 个字符串, 总长为 L , 问哪些串存在至少一种改变字母字典顺序的方案, 使其成为字典序最小的串。

算法讨论:

考虑 1 个字符串 s , 对于任意一个字符串 t , 如果令 $len=Lcp(s,t)$, 那么实际上的约束条件是 $s[1: len+1] < t[1: len+1]$ ($len < \text{length}(s) \&\& len < \text{length}(t)$), 我们对 n 个串建个 **trie**, 然后对于每个字符串我们在 **trie** 上走一遍, 取出所有的约束条件, 然后建图判环即可。

时间复杂度: $O(L+n*26*26)$

空间复杂度: $O(L*26)$

32. Codeforces 309D Tennis Rackets

题目大意:

给定一个正三角形, 每条边上有等距的 n 个点, 其中每条边两侧的 m 个点不能选择, 问你有多少种方案来选择其中 3 个点(每条边一个点), 使得它们构成一个钝角三角形。

算法讨论:

暴力枚举其中 2 条边上取的点, 另一条边上可取得范围是单调的。

时间复杂度: $O(n^2)$

空间复杂度: $O(1)$

33. GCJ Final 2009C Doubly-sorted Grid

题目大意:

一个长方形的网格每个格子里有一个小写英文字母。我们称这个网格是双排序(**doubly sorted**)的, 当每一行从左到右都是不下降的, 每一列从上到下也是不下降的。

给你一个部分格子填了英文字母的网格。你的任务是计算有多少种方式把剩下的格子填满, 使所得网格是双排序的。答案模 10007

算法讨论:

用 $f[i][s]$ 表示当前填到第 i 个字符, 已填部分的边界线(必定是从右上到左下的一段单调的折线)为 s 的方案数, 转移可以预处理, 接着 **dp** 即可。

时间复杂度: $O(26*r*C(r+c, r))$

空间复杂度: $O(r*2^{(r+c)})$

34. Usaco 2008 Nov Toys

题目大意:

有 D 天, 第 i 天需要已消毒的玩具 T_i 个, 新买一个已消毒玩具价格是 T_c , 消毒一个玩具可以花费 C_1 并等待 N_1 个晚上或花费 C_2 并等待 N_2 个晚上, 问满足所有需求的最小花费。

算法讨论:

费用流的做法是显然的, 但是需要卡常数。

我们设 $f(x)$ 表示一共新买 x 个玩具的情况下的最小花费, 那么这个时候我们可以一开始就买好 x 个, 然后这时候我们的决策可以直接贪心, 复杂度是 $O(d)$ 的。然后我们注意到 $f(x)$ 实际上是单峰的, 因为其随着 x 的增大, $f(x+1)-f(x)$ 是单调的, x 是可以三分的。

时间复杂度: $O(d \log d)$

空间复杂度: $O(d)$

35. Codeforces 273E Dima and Game

题目大意:

给 n 对整数 $(l[i], r[i])$ ($1 \leq l[i], r[i] \leq p$), 双方轮流操作:

将第 i 对数(满足 $r[i] - l[i] > 2$) 替换为

$(l[i] + \text{floor}((r[i] - l[i]) / 3), l[i] + 2 * \text{floor}((r[i] - l[i]) / 3))$

或 $(l[i], r[i] - \text{floor}((r[i] - l[i]) / 3))$

$\text{floor}(x)$ 表示向下取整。不能进行操作的玩家则输。

给定 n 和 p , 计算有多少种初始局面是先手必胜的。答案模 $1000000007 (10^9 + 7)$ 。

算法讨论:

注意到一个数对的 sg 值只和 $r - l$ 相关, 一开始计算出 sg 不同的长度的区间, 以及每种 sg 值一共有多少种数对(sg 值是 ≤ 2 的), 接着 dp 即可。

时间复杂度: $O(n + ?)$ 不同的 sg 的段数很少, 但是不知道如何计算 p 以内的段数

空间复杂度: $O(n + ?)$

36. Codeforces 243C Colorado Potato Beetle

题目大意:

有一个无限的矩阵, 一开始一个点位于格子 $(0, 0)$, 然后给出这个点的移动路线(分为 n 段, 每段是沿直线移动一定单位距离), 并把路线上经过的所有格子染色。现在有一个点从无穷远处出发, 不能经过已染色的格子, 问有多少格子不能到达。

算法讨论:

离散后暴力模拟一遍即可。

时间复杂度: $O(n^2)$

空间复杂度: $O(n^2)$

37. Codeforces 323C Two permutations

题目大意:

你有两个各包含 n 个元素的排列 p 和 q , 和 m 个由 l_1, r_1, l_2, r_2 组成的询问。每次询问在 p 中位置在 $[l_1, r_1]$, 在 q 中位置在 $[l_2, r_2]$ 中的数的数量。

算法讨论:

预处理 p 中每个位置的数在 q 中的位置序列 s , 然后询问等价于询问在 s 的 $[l_1, r_1]$ 中有多少个数属于 $[l_2, r_2]$, 可持久化线段树即可。

时间复杂度: $O((n + m) \log n)$

空间复杂度: $O(n \log n)$

38. Codeforces 248E Piglet's Birthday

题目大意:

有 n 个架子, 第 i 个架子上有 $a[i]$ 个蜜罐, 一开始都是满的, 有 q 次操作; 第 i 次会去第 u_i 个架子, 随机拿走 k_i 个蜜罐, 并将他们变成空的, 然后放到第 v_i 个架子上。询问每次操作后, 架子上所有蜜罐都被吃完的架子的期望个数是多少。

算法讨论:

暴力维护 $f[i][j]$ 表示第 i 个架子上有 j 个蜜罐是满的概率。

时间复杂度: $O(nka)$

空间复杂度: $O(na)$

39. Codeforces 264E Roadside Trees

题目大意:

在一条直线有 n 个位置可以种树, 自西向东标号 $1 \sim n$ 。每个月, 每棵树都会长高 1 米。每个月初, 你都会收到一个要求。要求有两种类型:

- 1、在位置 p 种一棵高度为 h ($h \leq 10$) 的树
- 2、砍掉从西向东数的第 x ($x \leq 10$) 棵树, 当这个位置的树被砍掉后, 倒下的树会占据这个位置, 之后这个位置不能再种树。

在做完这个要求后, 你需要求出当前树高的最长上升子序列。所以任何时刻树的高度互不相同。

算法讨论:

注意到 x 和 h 都很小, 因此种植操作和删除操作实际上影响的树都只有不超过 10 棵, 分 2 类暴力把这些树取出再重新插入即可。线段树维护。

时间复杂度: $O(mh \log n + mk \log n)$

空间复杂度: $O(n)$

40. Codeforces 264D Colorful Stones

题目大意:

有 2 个字符串 A 、 B (仅包含 RGB), 两个人分别站在两串的开头, 每次可以选择 RGB 中的一个, 然后每个人如果发现脚下是这个字符就前进。问一共有多少个状态是可达的。

算法讨论:

首先有一个结论 (x, y) 可达, 当且仅当 $A[1..x+1]$ 不是 $B[1..y]$ 的子序列, $B[1..y+1]$ 不是 $A[1..x]$ 的子序列, 且 $A[x] \neq B[y-1]$ 或 $A[x-1] \neq B[y]$ 。

然后暴力即可。

时间复杂度: $O(n)$

空间复杂度: $O(n)$

41. Codeforces 268D Wall Bars

题目大意:

问有多少方案满足:

- 1、我们定义单位长度, 建筑物中间那根管子高度为 n 。
- 2、在高度为 $1, 2 \dots n$ 的地方, 恰好有一根水平的横杆从中间的杆子连向四个方向中的某一个预先固定好的杆子上。
- 3、如果两根横杆的距离不超过 h , 且方向相同, 那么一个孩子可以从一个一根横杆爬到另一根上。在地上的孩子, 可以爬到任何一根高度在 $1-h$ 之间的横杆上。在 Manao 的建筑物上, 一个从地面出发的孩子至少能到达一根高度在 $n-h+1, n-h+2 \dots n$ 的横杆。

输出答案模 1000000009

算法讨论:

暴力 dp, 用 $f[c][i][j][k][t]$ 表示高度 c , 另外 3 个方向的杆子据当前高度 i, j, k, t 表示当前是否满足条件 3

时间复杂度: $O(n \cdot h^3)$

空间复杂度: $O(h^3)$

42. GCJ Final 2014D Paradox Sort

题目大意:

有 n 个蜡烛, 给出每对蜡烛间的偏好关系, 一开始只有 1 根蜡烛, 当面对 1 根新蜡烛时会保留已有蜡烛和新蜡烛中较喜好的那一根, 问是否存在一个排列使得最后能保留指定的某根蜡烛, 如果存在, 输出字典序最小的排列。

算法讨论:

如果相较于 B 更偏好 A , 那么就从 A 连边到 B 。当前局面有解当前仅当每个点都能被目标点访问到, 然后枚举排列的下一个位置是多少, 暴力验证。

时间复杂度: $O(n^4)$

空间复杂度: $O(n^2)$

43. Codeforces 280D k-Maximum Subsequence Sum

题目大意:

给定定整数序列 $a[1], [2], \dots, a[n]$, 你应该处理以下两种操作。

把 $a[i]$ 赋值为 val 。

询问从 $a[l \sim r]$ 取不超过 k 段的最大子段和。

算法讨论:

考虑费用流的做法, 实际上是每次取最大的子段和, 并把这段序列取负, 用线段树模拟费用流过程即可。

时间复杂度: $O(nk \log n)$

空间复杂度: $O(n)$

44. Codeforces 243D Cubes

题目大意:

给定 $n \times n$ 的方阵, 每个格子上堆有若干正方体, 问在一个无穷远的地方以给定向量观望可以看到多少个立方体。

算法讨论:

实际上要求的就是每个格子上最低的能被看到的位置, 我们可以按照距离观望方向的前后位置排序, 然后用线段树维护区间的最小值即可。

时间复杂度: $O(n \log n)$

空间复杂度: $O(n)$

45. Codeforces 342D Xenia and Dominoes

题目大意:

一个拼图是一个 $3 \times n$ 的桌子, 除掉一些禁止块, 并包含多米诺骨牌。一个拼图被称作合法当它符合以下条件:

每个多米诺骨牌覆盖正好两个非禁止块;

没有两个多米诺骨牌覆盖桌子上的同一块区域;

有且仅有一个非禁止块没被任何多米诺骨牌覆盖 (图中的圆点)。

一步移动是在保证拼图合法的情况下, 把一个多米诺骨牌移到空格里。横向的多米诺骨牌只能横向移动, 纵向的多米诺骨牌只能纵向移动。不能旋转。

问有多少种放骨牌的方案使得拼图至少能移动一次

算法讨论:

暴力枚举圆点能朝哪几个方向移动，容斥解决。计算方案使用状压 dp 。

时间复杂度: $O(64n)$

空间复杂度: $O(8n)$

46.Usaco 2009 Mar Cleaning Up

题目大意:

n 头奶牛排成一个序列，一共 m 种食物，每头奶牛需要一种食物，每一次你可以从序列的头取出若干奶牛，并付出这些奶牛所需不同食物种类的平方的代价，求使序列为空的最小代价。

算法讨论:

注意到如果每次都只取 1 头奶牛那么总代价是 n ，所以答案是小于等于 n 的，用 $dp[i]$ 表示序列的前 i 头奶牛所需最少代价，转移时只需要考虑使最后一段不同食物种类 $\leq \sqrt{n}$ 的最前位置即可。

时间复杂度: $O(n^{1.5})$

空间复杂度: $O(n)$

47.Usaco 2009 Jan Safe Travel

题目大意:

给定 n 个点、 m 条边的边有权无向图，对于每个点 i ，询问从点 1 出发不经过 1 到 i 的最短路（保证存在且唯一）上的最后一条边的最短路。

算法讨论:

先做出从点 1 出发到其他所有点的最短路树，设 $d[x]$ 表示 x 到 1 的最短路，询问时点 i 的最优解一定是经过了一条非树边 (u,v) 使得 u 在 i 的子树外， v 在 i 的子树内，这时候的解是 $d[u]+e(u,v)+d[v]-d[i]$ ，用可并堆/并查集维护最小值即可。

时间复杂度: $O((n+m)\log m)/O(n+m)$

空间复杂度: $O(n+m)$

48.Usaco 2008 Dec Best Cow Line

题目大意:

给定长度为 n 的字符串 A ，字符串 B 初始为空，你每次可以从 A 的头或尾取出 1 个字符并放到 B 的尾，求字典序最小的 B 。

算法讨论:

每一步取头或者取尾取决于当前的串 A 与它翻转后的串 A' 的字典序大小，二分+hash 比较即可。

时间复杂度: $O(n\log n)$

空间复杂度: $O(n)$

49.Usaco 2012 Mar Cows in a Skyscraper

题目大意:

电梯最大承重 W ， n 只奶牛，第 i 只奶牛体重 C_i 。求最少坐多少次电梯才能运完所有的奶牛。

算法讨论:

状压 dp , 用 $f[s], g[s]$ 表示当前已运完的奶牛集合为 s , 至少要用 $f[s]$ 次电梯, 最后一次的电梯至少要载总重 $g[s]$ 的奶牛, 转移时直接枚举接下来加入的奶牛即可。

时间复杂度: $O(n \cdot 2^n)$

空间复杂度: $O(2^n)$

50. Usaco 2010 Open Triangle Counting

题目大意:

给出 n 个笛卡尔坐标系上的整点, 统计有多少三角形包含原点 $(0,0)$ 。

算法讨论:

考虑算出有多少个三角形不包含原点, 对于每个不包含原点的三角形, 我们枚举其中极角序最小的点, 那么另外 2 个点都要在这个点与原点所在直线的同一侧, 用一个单调指针计算即可。

时间复杂度: $O(n \log n)$

空间复杂度: $O(n)$

51. Codeforces 269D Maximum Waterfall

题目大意:

平面上有 n 条平行于 x 轴的线段, 任意 2 条线段没有公共点, 第 i 条线段能流向第 j 条当且仅当:

1. 二者的水平位置存在相交部分
2. 第 j 块板在第 i 块板下方
3. 不存在 k , 且 (i,k) 与 (k,j) 均满足以上两个条件。

从 i 到 j 的流量为二者的水平相交部分的长度。

如果水流到了一块水平板上 (除了墙的底部), 水会流向恰好一个更低的水平板。整个瀑布的水流量被定义为水流路径上每连续的两块水平板间的水平相交部分长度的最小值。

求可能的最大水流。

算法讨论:

用 $f[i]$ 表示从 i 出发的最大水流, 按高度从小到大顺序 dp , 用一个 set 维护可能的转移的区间, 暴力转移即可。

时间复杂度: $O(n \log n)$

空间复杂度: $O(n)$

52. Codeforces 241E Flights

题目大意:

给定 n 个点 m 条边的有向图, 边权为 1, 让你把某些边的权值变成 2 使得从 1 到 n 的所有路线等长。

算法讨论:

转化成查分约束模型跑 $spfa$ 即可。

时间复杂度: $O(k(n+m))$

空间复杂度: $O(n+m)$

53. Gcj Final 2009D Wi-fi Towers

题目大意:

平面上有 n 个点, 每个点有覆盖的范围(以这个点为圆心的一个圆), 升级点 i 可以得到 ci 的收益(可能为负), 如果点 i 升级, 那么所有点 i 覆盖范围内的点也必须升级。问最大收益。

算法讨论:

这题是裸的最小割模型, 用 S 、 T 表示源点和汇点, 如果一个点收益为正那么预先将其计入答案并从 S 向其连容量为收益的边, 否则将其向 T 连边, 如果 i 能覆盖 j 就从 i 向 j 连 inf 边。求最小割即可。可以通过优化将边数优化到 $O(n)$ 。

54. Codeforces 241D Numbers

题目大意:

给定 n 的一个排列, 要求删除一些数, 使得结果序列满足:

1. 序列非空
2. 序列中所有数异或和等于 0
3. 把所有数按十进制从前往后依次无间隔地写在一行形成一个大的十进制数, 这个数将会被 p 整除

算法讨论:

当 $n > 31$ 时如果我们只考虑 $1 \sim 31$ 这些数, 满足条件 1 和 2 的序列有大约 $6 \cdot 10^7$ 个, 当有解时, 我们能找到解的概率是很接近 1 的, 直接 dp 即可。

时间复杂度: $O(32^2 \cdot p)$

空间复杂度: $O(32^2 \cdot p)$

55. Codeforces 319E Ping-Pong

题目大意:

在这个问题中, 每个时刻有一个区间的集合。每次可以从集合中的区间 (a, b) 移动到另一个满足 $c < a < d$ 或者 $c < b < d$ 的区间 (c, d) 。需要支持插入一个区间以及询问是否有一种从区间 x 到区间 y 的移动方案。保证每次插入的区间长度严格大于已插入的所有区间。

算法讨论:

从 i 能到 j 的前提是 j 包含 i 或者 j 与 i 相交。如果只考虑相交, 我们要知道的就是每个区间向左向右最长延伸多少, 用线段树和并查集维护互相可达的区间。

而询问 x 能否到 y 时我们只要判断 x, y 是否在同一集合以及 x 所在集合所对应的大线段是否能到达 y 即可。

时间复杂度: $O(n \log n)$

空间复杂度: $O(n \log n)$

56. Codeforces 306D Polygon

题目大意:

要求构造一个每个角角度都相同、每条边长度都不相同、边数给定的凸多边形。

算法讨论:

先取一个点做起点, 每次走长度 len 然后旋转固定的角度, 并把 len 小幅增

加，做完 $n-1$ 个点后，最后一个点取交点得到。

时间复杂度: $O(n)$

空间复杂度: $O(n)$

57.Usaco 2010 Mar StarCowraft

题目大意:

游戏中有 3 种单位，强度分别为 s_1 、 s_2 、 s_3 (强度未知)，一支军队的强度是军队中所有单位的强度和，军队交战时强度高的将获胜，强度相等时将会随机出现一个胜者。现在给出 n 次已知结果的战斗，询问另外 m 次战斗的胜负情况。

算法讨论:

已知结果的战斗对应若干个线性约束，询问实际上询问某个线性表达式的最大和最小值。由于是二维的线性规划，直接使用半平面交即可。

时间复杂度: $O(n^2+mn)$

空间复杂度: $O(n)$

58.Usaco 2013 Mar Hill Walk

题目大意:

有 n 条斜率为正数的线段，第 1 条线段左端点为 $(0,0)$ ，线段不会相交或接触，一个点从 $(0,0)$ 出发，每次走到当前线段的尾端，并移动到当前位置向下遇到的第一条线段上，知道下方没有线段为止。求在多少条线段上走过了正数长度。

算法讨论:

由于线段互不有交，用一条平行于 y 轴的扫描线维护每次下落遇到的第一条线段即可，用 `set` 维护。

时间复杂度: $O(n\log n)$

空间复杂度: $O(n)$

59.Usaco 2013 Open

题目大意:

有 N 头奶牛排成一排，标号为 $1..N$ 。照片 i 包含了从 a_i 到 b_i 的奶牛，每张照片中恰有一头奶牛有斑点。问至多有多少头奶牛有斑点。

算法讨论:

用 $f[i]$ 表示第 i 头奶牛有斑点的情况下前 i 头奶牛最多有多少头有斑点，转移时 $f[i]=f[j]+1$ 要求 j 不被 i 覆盖 i 的区间覆盖且不存在一个区间 $[s,t]$ 使得 $s>j$ 且 $t<i$ ，然后单调队列优化。

时间复杂度: $O(n)$

空间复杂度: $O(n)$

60.Codeforces 258D Little Elephant and Broken Sorting

题目大意:

给定 $1\sim n$ 的排列， m 次操作，每次有 0.5 的概率交换 2 个位置上的数，问最后逆序对的期望对数。

算法讨论:

用 $f[i][j]$ 表示第 i 个位置上的数大于第 j 个位置上的数的概率，暴力维护就行了。

时间复杂度: $O(n^2+nm)$

空间复杂度: $O(n^2)$

61.Codeforces 293B Distinct Paths

题目大意:

有一个 $n*m$ 的木板, 一些块已经被涂上给出的 k 种颜色中的一种。你需要把每个没涂色的块涂色使得从左上角到右下角的每条路径都不会经过两个颜色一样的块。路径只能向右或向下走。求方案数模 10^9+7

算法讨论:

搜索加剪枝。

62.Codeforces 253E Printer

题目大意:

有 n 个任务, 第 i 个任务有接收到的时间 t_i 、所需时间 s_i 、优先级 p_i 。每个单位时间内会选择优先级最高的任务执行。现在有一个任务 w 优先级未知, 但是知道它的完成时间 T , 求该任务的优先级以及每个任务的完成时间 (给出一组解)。

算法讨论:

如果已知优先级, 我们可以用堆在 $O(n\log n)$ 的时间内模拟得出每个任务的完成时间, 不难发现完成时间是随优先级单调变化的, 所以可以二分。复杂度是 $O(n\log^2 n)$ 的。

但是我们可以不用二分, 先将所求优先级赋为最低, 求出每个任务在时间 $[t_w, T]$ 之间所占的时间, 然后把所有任务按优先级排序扫描一遍即可求得所求优先级。

时间复杂度: $O(n\log n)$

空间复杂度: $O(n)$

63. Codeforces 274C The last Hole

题目大意:

平面上有 n 个圆, 一开始所有圆半径为 0 , 在时刻 t , 所有圆半径为 t , 一个洞定义为被几个圆的面积并环绕形成的封闭部分, 求最后一个洞消失的时刻。

算法讨论:

当 3 个圆心构成锐角三角形或当 4 个圆心构成矩形时才会有最后的洞, 暴力枚举即可。

时间复杂度: $O(n^4)$

空间复杂度: $O(n)$

64.Usaco 2007 Jan Cow Schul

题目大意:

第 i 次考试得分 T_i , 满分为 P_i 。在计算总成绩 G 前, 将把分数率 $F_i=T_i/P_i$ 最低的 d 份试卷去掉, 总成绩 $G=(\sum T_i)/(\sum P_i)$ 。

求出所有的 d , 使得去掉 d 份试卷后 G 可以比上述算出来的更高。

算法讨论:

我们把考试按 F 排序, 对于每个 d , 我们需要知道的实际上对于上述算法

算出的 G ，是否存在把第 i 次成绩换成第 j 次成绩后 G 更大($i \leq d, j > d$)，也就是是否存在 $P_i - G \cdot T_i < P_j - G \cdot T_j$ ，那么我们只要分别求出前 d 次成绩中 $P_i - G \cdot T_i$ 的最小值和第 $d+1$ 次以后的成绩中 $P_i - G \cdot T_i$ 的最大值即可。用 `set` 维护。

时间复杂度: $O(n \log n)$

空间复杂度: $O(n)$

65.Gcj Final 2011A Runs

题目大意:

有一个由小写字母 “a” - “z” 组成的字符串。每一个极大连续相同的子序列被称为一个 `run`。

问将给定的字符串重新排列，有多少种不同的结果使得它和原字符串 `run` 的个数相同。模 1000003。给定串的 `run` 为 m 个。 m 不超过 100。

算法讨论:

用 $f[i][j]$ 表示已加入了前 i 种字符，当前字符串有 j 个 `run` 的方案数，用组合数转移即可。

时间复杂度: $O(26m^2)$

空间复杂度: $O(26m)$

66.Codeforces 316D PE lesson

题目大意:

你有一个长度为 n 的序列，每个位置上依次为 $1 \sim n$ ，你每次可以交换 2 个不同位置上的数，给出一个数组 $a[i]$ 表示第 i 个位置上的数最多被交换 $a[i]$ 次。问最终不同序列的个数模 10^9+7

算法讨论:

方案数等价于把这 n 个位置分成若干个集合，每个集合内至多有 2 个 1 的方案数。证明可以通过每次增加一个位置来证。然后就直接 `dp` 即可。

时间复杂度: $O(n)$

空间复杂度: $O(n)$

67.Codeforces 316F Suns and Rays

题目大意:

给定一副图片，由太阳以及纯色背景组成。太阳是带有光线的可以任意旋转的椭圆。太阳射线是一条连接在椭圆边界上的线段。计算太阳的数量和计算每个太阳射线的数量。

算法讨论:

扫描整个图片得到每个太阳，太阳数直接用连通块个数即可，然后对于每个太阳用一定宽度的方块把中间的椭圆大致抹掉，然后统计大小超过一定的连通块个数即认为是射线数目。

68.Codeforces 261D Maxim and Increasing Subsequence

题目大意:

给定长度为 n 的数列 b ，数列 a 是 t 个数列 b 拼接的结果，求 a 的最长上升子序列。

算法讨论:

先将序列 b 离散, 设 b 有 tot 种不同数值, 如果 $t \geq \text{tot}$, 那么答案显然是 tot 。否则, 我们令 $f[i]$ 表示以第 i 小的权值为尾的最长上升子序列长度, 然后暴力 dp 即可。

时间复杂度: $O(n \cdot \max b)$

空间复杂度: $O(\max b)$

69.Codeforces 325E The Red Button

题目大意:

n 个点的图, 点编号 $0 \sim n-1$, 第 i 个点向 $2i \bmod n$ 和 $2i+1 \bmod n$ 各有一条有向边, 求一条哈密顿回路。

算法讨论:

首先 n 为奇数一定无解。因为 0 的入边来自 0 和 $(n-1)/2$, $n-1$ 的入边来自 $n-1$ 和 $(n-1)/2$, 而 $n-1$ 只有一个后继。

然后注意到 x 和 $x+(n/2)$ 的出边是相同的, 我们先给每个点任意指定后继, 然后对于 2 个回路, 我们找到其中出边相同的点, 交换后继即可将 2 条回路合并。

时间复杂度: $O(n)$

空间复杂度: $O(n)$

70.Usaco 2013 Open Figure Eight

题目大意:

给定 $n \times n$ 的矩阵, 每个位置是完好或破损的, 一个“8”定义如下:

数字 8 由上下两个矩形构成, 上下两个矩形都满足至少有一个单元格在矩形内部, 顶部的矩形的底边必须为底部矩形顶边的子集。且只能刻在大理石完美无瑕的部分。

数字 8 的得分为上矩形和下矩形的面积的乘积, 求最大得分。

算法讨论:

枚举 8 下矩形的上边界, 通过预处理来计算面积即可。

时间复杂度: $O(n^3)$

空间复杂度: $O(n^3)$

71.Codeforces 249D Donkey and Stars

题目大意:

在坐标系第一象限上有 n 个点, 一开始位于原点, 每次你可以到达位于一个夹角内的另外一个点。夹角是一开始给出的。问最多能移动几次。

算法讨论:

计算出描述夹角的 2 条射线所对应斜率过每个点在 y 轴的截距, 那么从点 i 能到点 j 等价于一个二维的偏序, 排序后 dp 即可。

时间复杂度: $O(n \log n)$

空间复杂度: $O(n)$

72.Codeforces 235D Graph Game

题目大意:

给定一棵环加外向树, 问在这个图上做随机选点的树分治算法的期望复杂度。

算法讨论:

我们考虑在选取点 i 使点 j 做出的贡献即点 j 与点 i 仍连通的概率,如果 i 、 j 在同一棵外向树内概率就是 $1/\text{dist}(i,j)$, 否则就是有 2 条路径, 那么这时概率就是 $1/\text{dist1}(i,j)+1/\text{dist2}(i,j)-1/(\text{dep}(i)+\text{dep}(j)+\text{len_circle})$, $\text{dep}(i)$ 指 i 在其所属外向树中的深度。暴力计算即可。

时间复杂度: $O(n^2)$

空间复杂度: $O(n)$

73.Usaco 2011 Jan Bottleneck

题目大意:

n 个点的有向树, 点 i 上有 C_i 的奶牛, 向其父节点 P_i 连了一条有向边, 每个单位时间内, 不超过 M_i 的奶牛能从 i 到 P_i (认为瞬间完成), 给出若干个 T , 询问在 T 时刻前最多有多少奶牛能到达根。

算法讨论:

把询问排序, 按时间段模拟, 实际上最多能到达的奶牛是若干段形如 $kT+b$ 的一次函数, 用堆维护每个点与父亲连边不会再满的最早时刻, 维护一棵等价的树即可。

时间复杂度: $O(n \log n)$

空间复杂度: $O(n)$

74.Gcj Final 2009B Min Perimeter

题目大意:

给你一个整数坐标的点集, 询问点集中最小的三角形周长是多少。退化的三角形也是允许的(面积为 0)。

算法讨论:

分治, 每次按 x 坐标把点分成 2 个集合, 分别求解后, 对于跨分界线的答案按 y 坐标排序, 维护一个正方形内的点集并暴力求答案。

时间复杂度: $O(n \log^2 n)$

空间复杂度: $O(n)$

75.Codeforces 267C Berland Traffic

题目大意:

构造给定无向图的最大流, 使从 S 到 T 的每条路径流量相同。

算法讨论:

设每个点的势能并根据流量平衡进行消元。

时间复杂度: $O(n^3)$

空间复杂度: $O(n^2)$

76.Codeforces 329E Evil

题目大意:

给定 n 个平面上的点, 两点距离为曼哈顿距离, 求最长哈密顿回路长度。

算法讨论:

首先我们可以计算出答案的一个上界, 把两维坐标分别排序, 然后尽量让坐标大的取正号。我们令 x_m, y_m 分别为 x, y 的中位数, 那么通过分类讨论可以得

出, 这个上界只在把坐标轴移至 (x_m, y_m) 后每个象限均有点且 n 为偶数或者有点位于 (x_m, y_m) 时无法取到, 这时我们需要调整方案。我们只需要尝试中位数左右 x 的差和 y 的差即可。

时间复杂度: $O(n \log n)$

空间复杂度: $O(n)$

77.Codeforces 343E Pumping Stations

题目大意:

给定一个无向图 G , 边有容量, 求一个哈密顿路径, 使得路径长最大, 两点间的边权为 G 中两点间的最大流。

算法讨论:

G 的 Gomory_Hu 树是这样一棵无向树, 使得 G 中 2 点 i, j 的最大流(最小割)为树上 2 点间边权的最小值, 我们通过 n 次网络流建出这棵树, 然后算出其边权和即可。

78.Codeforces 293D Ksusha and Square

题目大意:

给定一个面积不为 0 的凸多边形。随机选择两个不同的格点(格点可以在凸多边形内部或者边界上面), 并以它们之间的连线为对角线做一个正方形, 那么这个正方形的面积期望是多少? 坐标范围 $-10^6 \leq x \leq 10^6$

算法讨论:

我们要算的面积等价于距离的平方/2, 所以我们只要分别算出两维坐标差的平方的期望值即可。以 x 为例, 统计出每个 x 坐标有多少个点, 然后维护式子中每项的期望即可。

时空复杂度: $O(n + \text{坐标范围})$

79.Shaass and Paint Robot

题目大意:

给定 $n*m$ 的网格。机器人从边界上的点出发, 沿 45 度角移动, 碰到角就反向, 碰到边就反射, 机器人到达过的格子将染成黑色, 问能否将网格染成黑白相间, 如果能, 需要几步。

算法讨论:

可以证明得到一个结论: 网格黑白相间等价于边界黑白相间。因此我们只要模拟直到边界上所有该染过的点都染过即可。

时间复杂度: $O(n+m)$

空间复杂度: $O(n+m)$

80.Codeforces 309B Context Advertising

题目大意:

你得到了一个包含 n 个单词的文本。一个正式的广告标语有恰好 r 行, 每一行最多包含 c 个字符。问最多能把多少个连续单词写入广告。

算法讨论:

扫描一遍得到每个单词为行首最多能放到的位置, 倍增算出每个单词为广告的起始能放到的最后位置。

时间复杂度: $O(n \log n)$

空间复杂度: $O(n)$

81.Gcj Final 2014C Symmetric Trees

题目大意:

给定一棵树, 点有颜色, 判断能否将这棵树在坐标系上摆放使边只在顶点有交, 且关于一条轴对称。

算法讨论:

显然有一个点在轴上或着一条边过轴, 那么我们在每条边中加 1 个点, 就只用考虑有点在轴上的情况了。然后枚举轴上的点, 树的 hash 解决。

时间复杂度: $O(n \log n)$

空间复杂度: $O(n)$

82.Codeforces 301E Yaroslav and Arrangements

题目大意:

良好序列满足相邻两项以及头尾两项差为 1 且第 1 项是最小值。

优秀序列满足其元素单调不降且将其重排能得到至少 1 个至多 k 个良好序列。求长度不超过 n、最大值不超过 m 的优秀序列个数。

算法讨论:

dp。用 $f[i][j][k][1]$ 表示从小到大用了 i 种数值, 第 i 种数值用了 j 个, 序列长度为 k, 能排出 1 个良好序列的方案数。转移时枚举当前数值取了多少个即可。

时间复杂度: $O(n^5)$

空间复杂度: $O(n^4)$

83.Codeforces 241F Race

题目大意:

给一个方阵地图, 每个位置是障碍物、小写字母代表的路口或数字代表的路。其中数字同时表示从该格到相邻非障碍格的耗时, 所有数字构成的连通块只会是一个宽为 1 的矩形。给出起点、终点和途径路口。求时刻 k 所处的位置。

算法讨论:

模拟即可。

84.Codeforces 332E Binary Key

题目大意:

给定字符串 p 和字符串 s, 求字典序最小的长度为 k 的 01 字符串 q 使得 s 按 q 构造出的长度与 p 相同的串是 p。

构造方法是将 s 重复足够多次形成串 A, 并将 q 重复足够多次形成串 B, 从左往右逐位判断, 如果 B 的该位为 1, 那么答案串末尾就加入 A 该位上的字符。

算法讨论:

枚举每个周期中有多少个 1, 然后贪心即可。

时间复杂度: $O(\text{len}(s) * \text{len}(p))$

空间复杂度: $O(\text{len}(s) + \text{len}(p) + k)$

85.Codeforces 346E Doodle Jump

题目大意:

T 组询问。对于每组询问，有 n 个平台，第 x 个平台的高度是 $ax \bmod p$ ，其中 a 和 p 是互质的正整数。Doodler 最大可能跳的高度是 h 。问能否跳到最高的平台。

算法讨论:

考虑如何求出所有平台间的最大间距。如果 $a*n < p$ 那么间距显然是 a 。否则我们考虑一个函数 $\text{find}(a, n, p)$ ，表示每次增量 a ，最高跳板为 p ，除最高跳板外有 n 个跳板。当 $a*n < p$ 时 $\text{find}(a, n, p) = \max(a, p - a*n)$ 。进行一些讨论可以得出，如果设 $k = a*n/p$ ，那么当 $a*n \bmod p < a*(p/a - 1)$ 时，有 $\text{find}(a, n, p) = \text{find}(\min(a - (p \bmod a), p \bmod a), k - 1, a)$ ，

否则有 $\text{find}(a, n, p) = \text{find}(\min(a - (p \bmod a), p \bmod a), k, a)$ 。

时间复杂度: $O(T \log p)$

86.Codeforces 251D Two Sets

题目大意:

给定 n 个数，将其分为 2 个集合 $S1$ 、 $S2$ ，两个集合内元素异或和分别为 $X1$ 、 $X2$ ，在最大化 $X1+X2$ 的情况下让 $X1$ 最小，输出方案。

算法讨论:

我们算出 n 个数的异或和 X ，如果 X 的某一位是 1，那么 $X1$ 的该位并不影响答案，否则我们应该尽量让该位为 1，那么我们先保证 $X1+X2$ 最大，从高位开始枚举 $X1$ ，如果 X 该位为 0，我们就尝试让 $X1$ 该位为 1，并高斯消元判断是否有解。

接下来我们最小化 $X1$ ，从高往低枚举 $X1$ 未确定的每一位即可。

时间复杂度: $O(n * 60^2)$

空间复杂度: $O(n * 60)$

87.Codeforces 341E Candies Game

题目大意:

给定 n 个数，每次可以取其中 2 个数 x 、 y ，执行 $y = y - x$ 、 $x = 2x$ 。要求使得最终恰有 2 数不为 0。

算法讨论:

如果至少有 3 个数不为 0，那么我们可以用这 3 个数互相操作使其中至少 1 个变成 0。我们设 3 个数为 $a > b > c > 0$ ，那么我们每次通过下面的操作使得 b 变成 $b \bmod c$ 。我们要做的事从 b 中移出 b/c 个 c ，那么我们用类似倍增的方法，如果当前 b/c 是奇数，我们从 b 中扣掉 c 使 c 翻倍，否则我们从 a 中扣掉 c 使 c 翻倍即可。

88.Codeforces 286E Ladies's Shop

题目大意:

n 个包，第 i 个包可以装恰好 a_i 的重量，让你构造一个物品重量的集合，使得每个包都能被使用，且对于任意个物品，如果其重量和不超过 M ，就要有一个包可以装下这些物品。并最小化集合的重量。

算法讨论:

注意到如果有一个 a_i 的倍数不超过 M 且没有这个重量的包，那么就无解。否则我们只要对每个不能被其他 a_i 的非负倍的和表示的 a_i 添加一个物品即可。通过 FFT 加速即可。

时间复杂度: $O(m \log m)$

空间复杂度: $O(m)$

89.Codeforces 360D Levko and Sets

题目大意:

给出 $a_1 \sim a_n, b_1 \sim b_m$, 由此生成 n 个集合, 第 i 个集合最开始只有 1 , 然后对于每个集合内的元素 $s, s * a_i^{b_j} \bmod p (j=1 \sim m)$ 也在集合中, 求这 n 个集合并的大小。

算法讨论:

设 $T = \gcd(p-1, b_1, b_2, b_3, \dots, b_m)$, 那么第 i 个集合中包含的就是所有的 $a_i^{kT} \bmod p$, 同时也可以表示成所有的 $g^{(k * \gcd(e_i, p-1))} \bmod p$, 其中 g 是 p 的原根, 且 $g^{e_i} \bmod p = a_i^T \bmod p$, 接下来可以通过容斥解决。

90.Codeforces 305D Olya and Graph

题目大意:

给定一个有向图, 问有多少种加边方案能使得其满足:

从点 i 出发, 可以到达点 $i+1 \sim n$ 。

任意从 u 到 v 的有向边满足不等式: $u < v$ 。无重边。

对于一对点 $i, j (i < j)$, 若 $j-i \leq k$, 那么从 i 到 j 的最短距离等于 $j-i$ 。

若 $j-i > k$, 那么从 i 到 j 的最短距离等于 $j-i$ 或 $j-i-k$ 。

算法讨论:

首先最终图上一定要有 $n-1$ 条编号相邻点之间的边才能保证前 2 个条件。除此以外就只能有从 i 到 $i+k+1$ 的边, 且不能有 2 条这类边覆盖的范围不相交。然后就枚举第一条这类边的位置, 然后乘上 2 的若干次方即可。

时间复杂度: $O(n)$

空间复杂度: $O(n)$

91.Codeforces 332D Theft of Blueprints

题目大意:

给出一个 n 个点的带权无向图, 满足对于任意一个大小为 k 的顶点集合 S , 恰好有一个点与 S 每一个点都有边。令这个点为 $v(S)$, 并且对 S 进行操作的代价是 S 中每个点与 $v(S)$ 的边权之和。现在求对于一个大小为 k 的子集操作代价的期望。

算法讨论:

当 $k \geq 3$ 时, 这张图一定是一个 $k+1$ 个点的完全图。因此读入后直接计算即可。 $k \leq 2$ 时也可以类似计算。

92.Gcj Final 2010C Candy Store

题目大意:

多组询问。 k 个顾客, 每个顾客要不超过 C 个糖果, 问你需要预先至少打包多少包糖果, 使得无论如何都可以满足每个顾客的要求。

算法讨论:

方法是只要当前的糖果总数不到 $k \cdot C$, 就继续打一包当前 $\text{floor}(\text{糖果总数} / (k+1))$ 的糖果.....证明还是略吧.....

93.Codeforces 338E Optimize!

题目大意:

给定序列 A、B, 长度分别为 n、m, 对 A 的每个长为 m 的连续子序列询问是否存在一种 B 的排列, 使得 A 的每个位置上的数和 B 的排列的对应位置的数的和均不小于 H

算法讨论:

把 B 变成 H-B, 问题变成询问是否存在一种对应关系, 使得 A 的每个位置上的数不小于 B 对应位置上的数, 那么我们就用线段树按权值大小顺序维护一个 +1/-1 的序列, 在 B 对应位置-1, A 的对应位置+1, 然后判断整个序列的最大后缀和是否小于 0 即可。

时间复杂度: $O((n+m)\log n)$

空间复杂度: $O(n+m)$

94.Codeforces 309E Sheep

题目大意:

给定 n 个区间, 两个区间之间有边当前仅当他们有公共点。要求构造一个排列使得有边的区间在序列中的最大距离最小。

算法讨论:

我们可以二分答案, 然后对于限制 len, 我们逐步地贪心给出方案。从左往右构造这个排列, 对于已构造的部分, 所有未使用的区间都有一个可以位置的前缀可以放, 然后我们就找到一个位置 i, 使得前 i 个位置恰要放满, 然后取这些区间中右端点最小的即可。

时间复杂度: $O(n^2 \log n)$

空间复杂度: $O(n^2)$

95.Codeforces 311C Fetch the Treasure

题目大意:

有 h 个位置, 依次标号为 1~h, 其中 n 个位置有宝藏, 第 i 个宝藏位于 $a[i]$, 价值为 $c[i]$ 。有一个集合 S, 一开始 S 中只有一个数 k。能到达的位置是所有的 $1 + \sum(v_i \cdot s_i, s_i \text{ 属于集合 } S \text{ 且 } v_i \geq 0)$ 。

有 m 个事件, 一共 3 类: 在 S 中加入一个数 x (最多只有 20 次)、让第 x 个宝藏的价值减少 y、查询当前能拿到的宝藏中价值最大的并拿走 (价值相同取较左边的)。

算法讨论:

注意到第一种操作很少, 我们可以每次暴力用最短路算出 f 数组, 其中 $f[i]$ 表示模 k 余 i 的数中能到的最小位置, 接下来用一个堆维护即可。

时间复杂度: $O(20k \log k + m \log n + n \log n)$

空间复杂度: $O(n+m+k)$

96.Codeforces 285E Positions in Permutations

题目大意:

一个排列 P 的第 i 个位置是完美的, 当前仅当 $\text{abs}(i-P[i])=1$ 。求长度为 n 且完美的位置数为 k 的排列数模 10^9+7 。

算法讨论:

用 $\text{dp}[i][j]$ 表示前 i 个数, 其中某 j 个位置是完美位置的方案数, 求出后 $\text{dp}[i][j]$ 后再乘上剩下元素的排列数得到至少 j 个位置完美的方案数, 之后容斥即可。

时间复杂度: $O(n^2)$

空间复杂度: $O(n^2)$

97.Codeforces 295D Greg and Caves

题目大意:

一个 $n*m$ 的网格, 问有多少种黑白染色方案满足: 有黑格的行构成连续一段。每行恰有 2 个或没有黑格, 称有黑色格子的行的两端黑格之间的白格为中间白格, 存在一个行, 从该行往上的每一个有黑格的行都满足其上面一行(如果有)的中间白格对应的该行位置均为白格, 从该行往下的每一个有黑格的行都满足其下面一行(如果有)的中间白格对应的该行位置均为白格。模 10^9+7

算法讨论:

用 $f[i][j]$ 表示前 i 行, 最后一行的中间白格长度为 j 、每一行的中间白格都要包含上一行的方案数, 最后枚举中间的行, 将上下的方案数相乘即可。

时间复杂度: $O(n*m)$

空间复杂度: $O(n*m)$

98.Gcj Final 2010A Letter Stamper

题目大意:

一个栈式的打字机, 支持下面三种操作:

(1) 把一个字母加到栈顶。

(2) 打印栈顶的字母。

(3) 删除栈顶字母。

给出你需要打印的字母序列(仅包含 **abc**), 你需要算出打印这个序列的最小操作数。

算法讨论:

可以证明最优解中栈中的元素只会是形如 **abcabcabc...** 的形式, 所以栈中的状态只有 $O(n)$ 种, 直接 **dp** 即可。

时间复杂度: $O(n^2)$

空间复杂度: $O(n)$

99.Codeforces 238E Meeting Her

题目大意:

n 个点 m 条边的有向图, 你要从点 **a** 到点 **b**, 有 k 条公交线路, 第 i 条从 **si** 到 **ti**, 公交车会随机选择一条从 **si** 到 **ti** 的最短路, 只要公交经过你就可以乘上公交, 问要到达目的地, 你最坏情况下至少要做几次公交。

算法讨论:

考虑最坏情况, 那么只有当公交必定会过当前所在点时我们才能坐上这条公

交线路，预处理出所有必经点，用 $f[i]$ 表示从 i 出发到 b 最坏情况下要坐多少次公交，然后 dp 即可。

时间复杂度： $O(n^4)$

空间复杂度： $O(n^2)$

100.Codeforces 257E Greedy Elevator

题目大意：

模拟一部电梯运做，每个时刻，这台电梯由上方等待人数+电梯内要到上方的人数-下方等待人数-电梯内要到下方的人数的值决定往上还是往下，如果这个值非负则往上，否则往下。给出每个人开始等待的时间、等待的楼层和要去的楼层，求每个人到达的时间。

算法讨论：

用堆优化模拟的过程即可。

时间复杂度： $O(n \log n)$

空间复杂度： $O(n)$
