2019牛客暑期多校训练营 第五场

dreamoon

A – digits 2

- 送大家温暖的签到题。
- 本题对于在范围内的 n 一定有解。
- 相信最简单的构造方法是把数字 n 当作字符串，输出 n 个头尾连接的字符串 n 就是答案。

- 此题 idea 是从一个经典问题而来：找到满足本题条件的最小的正整数解。没见过的人可以想想这个经典问题怎么做

牛客竞赛
AC.NOWCODER.COM

B – generator 1

- 其实就只是个没有额外技巧**矩阵快速幂习题**，跟一般的快速幂差别在于：以 2 为单位倍增要改成以 10 为单位。不需要把十进位转成二进位。
- 指数的每个位数还是要用 2 进制倍增计算，否则可能会 TLE。

- 出这题是因为我小时候曾被Codeforces 某个不是用2 进制倍增，而是改用10 进制倍增就会很好写的题震慑到，所以这次找了个纯用2 进制倍增有点困难的题和大家分享

C – generator 2

- **BSGS (Baby Steps Giant Steps) 算法的基础练习题**
- **不懂 BSGS 的人先去网上了解一下。 。 。**
- **在 a 不等于 0 的时候，xi 和 xi+1 是一对一对应的，所以可以直接套用 BSGS 算法**
- **由于同一组 generator 下有多组询问，若每组询问都使用 O(sqrt(p)) 的时间去查询的话，总时间复杂度变为 O(Q sqrt(p))，应该会得到TLE。**
    - 每个询问应该要用O(sqrt(p/Q)) 的时间去查询，这么做的话预处理的时间复杂度以及 Q 组查询的时间复杂度都会变为O(sqrt(pQ))

D – generator 3

- 以下做法仅供参考，说不定有更厉害的做法
- 序列 x 和序列 y 从某个位置后开始会出现循环，但循环节不一样
- 记 x 的循环节长度为 x_L，y的循环节长度为 y_L
- 先举个实例：
  - n = 15, x 从位置 3 开始循环, x_L = 5, y 从位置 2 开始循环, y_L = 4,
  - x0, x1, x2 , (x3, x4, … x,7), (x3, x4, … x,7), (x3, x4
  - y0,y1, (y2, y3, y4, y5) , (y2, y3, y4, y5) , (y2, y3, y4, y5) , y2

牛客竞赛
AC.NOWCODER.COM

D – generator 3

- n = 15, x 从位置 3 开始循环, x_L = 5, y 从位置 2 开始循环, y_L = 4,
- x0, x1, x2 (x3, x4, x5, x6, x7) (x3, x4, x5, x6, x7), (x3, x4
- y0, y1,(y2, y3, y4, y5) (y2, y3, y4, y5),(y2, y3, y4, y5),(y2

- **首先，不在循环节的部分可以独立出来处理**
  - 把 (x0, y0), (x1, y1), (x2, y2) 三个点抽出来后剩下 (x3, x4, ... x,7) 和 (y3, y4, y5, y2) 这两个循环
- **把 x 循环从第一个数开始以间距 y_L 为单位重新排列**
  - x 重新排列成 (x3, x7, x6, x5, x4)
  - 每个 y_i 对应到的 x 中的数都是循环中连续的一段！

牛客竞赛 AC.NOWCODER.COM

D – generator 3

- **求凸包时，对于每个 y 值，只需要知道该值对应到的最大的 x 及最小的 x 值即可。所以我们若能快速球出一个环状区间内的最大最小值，就能塞选出少量的点坐标**
  - 这就是 RMQ 问题。

- **所以本作法是个 求循环节 + RMQ + 凸包 的结合体**

- **注意，若 xL 和 yL 的 gcd 不为 1, 序列 x 会被分成很多循环，请读者自行揣摩这样的 case**

牛客竞赛
AC.NOWCODER.COM

D – generator 3

- n = 15, x 从位置 3 开始循环, x_L = 5, y 从位置 2 开始循环, y_L = 4,
- x0, x1, x2 ,(x3, x4, x5, x6, x7),(x3, x4, x5, x6, x7), (x3, x4
- y0, y1,(y2, y3, y4, y5),(y2, y3, y4, y5),(y2, y3, y4, y5),(y2

- **首先，不在循环节的部分可以独立出来处理**
  - 把 (x0, y0), (x1, y1), (x2, y2) 三个点抽出来后剩下 (x3, x4, … x,7) 和 (y3, y4, y5, y2) 这两个循环
- **把 x 循环从第一个数开始以间距 y_L 为单位重新排列**
  - x 重新排列成 (x3, x7, x6, x5, x4)

牛客竞赛
AC.NOWCODER.COM

E – independent set 1

- 基础的位元状压 dp 题
- 我们可以用一个 n-bit 2 进制整数来表示一个点集，第 i 个 bit 是 1 就代表点集包含第 i 个点，若是 0 则不包含
- 每个点相邻的点也可以用一个 n-bit 2 进制整数表示，计做 c_i，若第 i 个点和第 j 个点相邻，c_i 的第 j 个 bit 是 1，否则是 0
- 记 x 的最低位且是 1 的 bit 的位置是 lb_x
- 令 dp[x] 代表点集 x 的最大独立集 size，那么我们能够根据点 lb_x 是否属于最大独立集来列出以下关系式：
  $dp[x] = max(dp[x - (1<<lb_x)], dp[x \& (\sim c_{lb\_x})] + 1)$ (使用 c 语言运算符)

牛客竞赛 AC.NOWCODER.COM

E – independent set 1

- 时间和空间复杂度均为$O(2^n)$
- 为了符合本题的空间复杂度限制，必须使用 8-bit 的容器来储存 dp 的值

牛客竞赛
AC.NOWCODER.COM

F – maximum clique 1

- 二分图最大独立集的练习题
- 最大团问题和最大独立集问题是互补的问题
- 两个相异的数至少两个 bit 不一样的否命题就是：恰一个 bit 不一样
- 可发现按照题目叙述所建的图的补图就是刚好是二分图
- 于是套用二分图最大独立集模板就把这题解决了

G – subsequence 1

- **这是基础的 dp 练习题**
- **s 的所有长度大于 m 且不以 0 开头的 subsequence 都大于 t，这部分可以枚举开头位置和子序列长度，套用组合公式计算。**
- **s 长度等于 m 的 subsequence 部分使用动态规划计算：**
  - 令 dp (i, j) 表式 s 长度为 j 的后缀有多少 subsequence 长度为 i 且值大于 t 长度为 i 的后缀
  - 可以根据 s 从后面数来第 j 个位数是否包含在 subsequence 内来列出 dp 式，剩余部分读者自行练习

H – subsequence 2

- **应该算是拓扑排序的基础练习题吧**
- **实际上若不是无解，可能答案就只有一种**
  - 记由左边数来第 i 个字符c 的位置为 $pos_{c,i}$，根据 Input ，我可以知道所有 $pos_{c,i}$ 的大小关系
- **若把任两个字符所在的位置的大小关系都建出来，边数太多了**
  - 我们可以只建立 Input 中的字符串的相邻两个字符的位置大小关系即可
- **建完图后就套用拓扑排序模板即可**

H – subsequence 2

- 也可以不用实际建出图来，使用拓扑排序 bfs 做法的概念，只要由左到右依序决定 hidden string 的每个字符是什么即可。

I – three points 1

- **计算几何初级题(但在这套题里应该算是偏难的了(主要是实作难))**
- **两个观察：**
  - 1. 若一个三角形能摆在一个矩形里，总是能经过平移使得三角形至少有一个顶点和矩形的顶点重叠，且三角形的顶点仍在矩形里
  - 2. 重叠了三角形的某个顶点和矩形的某个顶点后，我们可以把该重叠的点当旋转轴，旋转三角形，使得三角形有另一个点恰好在矩形的某个边上
- **于是我们可以枚举三角形的哪个顶点和举行的顶点重叠，以及三角形另一个位在矩形边上的点是哪个，总是能枚举到一个完全落在矩形里面的三角形摆放位置**

J – three points 2

- **以下做法仅供参考，说不定有更厉害的做法**
- **显然无解的情况包括 ：**
  - 1. a+b+c-max({a,b,c}) < max({a,b,c})
  - 2. (a + b + c) 不被 2 整除
- **令路径 XY, YZ, 和 XZ 的交点为 O，可算出 OX 长度为(a+b-c)/2, OY 长度为 (a+c-b)/2, OZ 长度为 (b+c-a)/2**
  - 之后我们把这第 i 组询问的三个值由小到大记为 u_i, v_i, w_i
- **对于 tree 上一个点 C，若想知道他能不能是 O 点，可以采用如下方法：**

J – three points 2

- **先考虑只需处理一组询问**
- **对于 tree 上一个点 C，若想知道他能不能是 O 点，可以采用如下方法：**
  - 令 N_C 为所有 C 相邻的点所构成的点集，f(x,y) 代表所有满足 x 是一个端点且通过 y 点的 path 中，最长的 path 长度
  - 取出多重集合(元素可重复) $\{f(C,x)|x \in N_C\}$ 最大的三个值，設小到大依序為 p_C, q_C, r_C
  - 若满足 $p_C \geq u, q_C \geq v, r_C \geq w$,那么点 C 就可以当作 O 点
  - 使用 dp + dfs 两次的技巧，可以 O(n) 算出所有的 p_C, q_C, r_C
  - 若只有一组询问，找到能当作 O 点的点后，就可以暴力找出 X,Y,Z 的位置了

J – three points 2

- **若有多组询问**
- **上页算法有两个地方不够快**
  - 1. 知道 O 点，找出 X, Y, Z 三点
  - 2. 对与每个询问，都快速找出 O 点

牛客竞赛
AC.NOWCODER.COM

J – three points 2

- **若有多组询问**
- **知道 O 点，找出 X, Y, Z 三点**
  - 可简化问题为：给定点 x 和点 y，快速找出在路径 xy 上且距离 x 为 d 的点
  - 求 LCA (最近共同祖先) 的一种常见算法：倍增法所用到的技巧即可解决此问题
  - 经过预处理后，倍增法可以用 O(log n) 的时间复杂度知道一个点往 root 的方向经过 d 条边后会到达哪个点
  - 所以先求出 x 和 y 的 LCA，称为 z，由于可知道 x, y, z 的深度，所以也能知道， O 点应该是在 x 到 root 的路径上还是 y 到 root 的路径上

牛客竞赛
AC.NOWCODER.COM

J – three points 2

- **若有多组询问**
- **对与每个询问，都快速找出 O 点**
    - 把 (u,v,w) 当成平面上权重为 w, 座标为 (u, v) 的点， p,q,r 同理，坐标为 (p,q) 权重为 r
    - 我们要做的事就是，对于每个询问所代表的点，找到它右上角且权重不小于它的点
    - 此问题可用 扫描线 + 树状数组 解决
    - 把所有点依序比较 (x坐标, y 坐标, 权重) 由大到小排序，若完全一样，p, q, r 要排在 u, v, w 前面
    - 树状数组 在这里的作用为：支持在位置 i 插入一个(权重, id) 的 pair，并且可以询问位置 1 ~ i 中，权重最大的 id
    - 遇到 p, q, r 时，就在位置 n-q 插入 (r, id)，遇到 u, v, w，就查询位置 1~ (n-v) 中是否最大权重>= w，于是我们可以快速找到每个询问的O点了。

牛客竞赛
AC.NOWCODER.COM

J – three points 2

- 虽然这题用想的不太难，但写起来有点复杂，整体难度可能是所有其他题总和 :P
- 谢谢大家捧场写我这场欢乐赛(>__<)

Thanks

牛客竞赛
AC.NOWCODER.COM