

“文远知行”杯 中山大学程序设计竞赛线上赛 解题报告

出题人: Zayin、PureWhite、kqp

2020 年 4 月 25 日

A. 红宝石之光

只需统计每个正方体有多少个面是露出来的即可。

对于上表面和下表面，只有每一列的最上面的积木和最下面的积木是露出上表面或下表面的。

对于前后左右四个表面，例如对于 i 行 j 列的积木，只有 $\max(a_{i,j} - a_{i,j-1}, 0)$ 个积木是露出左表面的。其他表面同理。

B. 开花宣言

$O(2^n)$ 的时间复杂度枚举每种原料选或不选即可。可使用 dfs 实现。

C. 向着梦想的一步

一个重要的定理是， $\gcd(a, a + d) = \gcd(a, d)$ 。大家可以尝试证明一下。

因此，对于一个等差数列而言，所有项的 gcd 即为 $\gcd(a_1, d)$ 。

注意一种特殊情况，当 $n = 1$ 时，答案为 a_1 。

D. 夜空似乎知晓一切

一种较简单的方法是，解方程解出交点 (x, y) ，然后判断交点是否落在两条线段上。（即是否有 $x \in [x_1, x_2]$ 、 $x \in [x_3, x_4]$ 、 $y \in [y_1, y_2]$ 、 $y \in [y_3, y_4]$ ）

当然，这需要特判很多情况，比如平行、斜率不存在、 x_1 与 x_2 的大小顺序，等等。

如果发现讨论很麻烦，不妨百度一下“快速排斥实验+跨立实验”，这是一种计算几何的通法，其基本思想是用向量来表示直线，从而避免各种除法运算。

E. 冬天给予的预感

容斥原理。

$O(2^m)$ 的时间复杂度枚举各小组是否重复，设一共重复了 x 个小组 y 个人，则剩下的 $n - y$ 个人可以随意组成剩下的 $n - x$ 个小组（设人数分别为 p_1, p_2, \dots, p_{n-x} ），于是给答案加上 $(-1)^x F(n - y, p_1, p_2, \dots, p_{n-x})$ ，其中 $F(n, p_1, p_2, \dots, p_m)$ 表示 n 个人组成人数分别为 p_1, p_2, \dots, p_m 的小组的方案数。

$F(n, p_1, p_2, \dots, p_m)$ 怎么求呢？设 cnt_x 表示人数为 x 的组的数量，那么

$$F(n, p_1, p_2, \dots, p_m) = \frac{n!}{\prod_{i=1}^m p_i! \prod_{i=1}^n cnt_i!}$$

证明：这相当于有 n 个人任意站成一排，然后前 p_1 个人分成第一组，接下来 p_2 个人分成第二组，再接下来 p_3 个人分成第三组……的方案数。 n 个人站成一排有 $n!$ 种方案，但每个组内的成员不应该有顺序，即第 i 个组会造成 $p_i!$ 倍的重复，因此 $n!$ 要除以 $\prod_{i=1}^m p_i!$ 。并且，大小相同的组也不应当有次序之分，即大小为 x 的组会造成 cnt_x 倍的重复，因此还要再除去 $\prod_{i=1}^n cnt_i!$ 。

本题的除法要使用逆元运算，即提示中所给的方法。

F. 欲于辉夜之城起舞

解法一：

维护两个堆，比中位数小的数放入一个大根堆 H_1 ，比中位数大的数放入一个小根堆 H_2 。

每新加一个数 x ，跟中位数比较，不妨设其比中位数小，若 x 大于 H_1 的堆顶，那么 x 成为新的中位数，原中位数压入 H_2 ；若 x 小于 H_1 的堆顶，那么弹出堆顶代替中位数，原中位数压入 H_2 ， x 压入 H_1 。 x 比中位数大的情况同理。

时间复杂度 $O(n \log n)$ ，出题人说没把这复杂度卡掉是他的失误。

解法二：

倒序处理问题，先将所有数加进来，再从后往前一个个删除。

构建一个链表 $1 \rightarrow 2 \rightarrow \dots \rightarrow n$ ，并算出中位数的位置 $w = \lceil \frac{n}{2} \rceil$ 。

每次删除一个数，在链表中删除它，并判断 w 是否需要左移一位或右移一位。

时间复杂度 $O(n)$ 。

G. 欢乐派对列车

对于每个人，我们可以算出其无障碍走出去所用的时间 p_i 。

注意到只有 p_i 小的人会挡住 p_i 大的人，因此将所有人按 p_i 从小到大排序。设排完序后第 i 个人出去用时为 t_i ，那么就有 $t_i = \max(t_{i-1} + 1, p_i)$ ，这涵盖了所有情况。

H. 自我控制

设 dis_i 表示从起点出发到达 i 号城市的最短时间。这个可以用 bfs 来求，当我们在处理第 x 号城市时，若 $dis_x < \min(a, b)$ ，则两种边都可以使用；若 $\min(a, b) \leq dis_x < \max(a, b)$ ，则只能使用一种边；若

$dis_x \geq \max(a, b)$ ，则该城市无法更新别的城市。