

北航第十一届程序设计竞赛现场决赛题解

北航 2015 ACM-ICPC 集训队

2015 年 12 月 20 日

金字塔的难题

- 尝试人数：108
- 通过人数：96

- TAG：贪心，二分。

金字塔的难题

由于每一行的关卡之间相互独立，我们就有一个非常简单的贪心选择策略，**对于每一行从大往小的选，负数不选。**

这样我们就可以将每行的数排序，然后**枚举每行选的个数**，贪心的从每行都选数，直到总和 $\geq k$ 即可。

也可以**二分答案**，然后贪心选，最后检查是否满足题设条件。

乱六一糟的数列

- 尝试人数：91
- 通过人数：56

- TAG：贪心。

乱六一糟的数列

如果只考虑交换操作，显然花费等于“1”占用“6”的位置（也等于“6”占用“1”的位置）的个数。

对于排序操作，相当于将一段区间内的数1放在前面，6放在后面。这样的操作可以由几次将“61”交换成“16”得来，也就相当于将某一个“1”前移。

那么为了使尽量多的位数对应，将每一个“1”向左移至最左边对应位置即可。

模式匹配

- 尝试人数：77
- 通过人数：42

- TAG：模拟。

模式匹配

因为 $s_0 = a_0 + \frac{1}{s_1}$;

由 $\frac{1}{s_1} < 1$, 要比较 s_0 , 很明显应该先比较 a_0 。

如果 a_0 不同, 两个大小就比出来了。如果相同那就继续比较 $\frac{1}{s_1}$ 。

故显然两个数相等当且仅当 a 完全相等。

更大公约数

- 尝试人数：46
- 通过人数：38

- TAG：暴力。

更大公约数

考虑一维情况，可以预处理出前缀gcd和后缀gcd，快速算出删除某个元素的答案。

二维情况，可以分别处理出四个方向的上前缀和，上后缀和，下前缀和，下后缀和，快速计算删除某一行某一列的答案。也可以将矩阵压缩为一维来考虑。

验证码

- 尝试人数：62
- 通过人数：35

- TAG：模拟。

验证码

先判断一下 x,y 坐标是不是有且只有3种，而且中间的坐标是两边坐标的中点。

再把点重标号，看看应该有的线段是不是都有了即可。

- 尝试人数：13
- 通过人数：6

- TAG：数据结构。

$n = 5000?$

希望得到 $O(n^2)$ 的算法。

$O(n^2 \log n)$ 的算法太多了。

- $[l, r] \rightarrow [l, r + 1]$ 怎么不用带log的数据结构维护中位数?
- 反过来思考, 删除比插入简单。
- 固定区间左端点 l , 依次计算 $r = n, n - 1 \dots l$ 的答案。
- 用链表把排好序的 $a[l \dots r]$ 串起来, 考虑删去某个数后中位数怎么移动。
- 排序? $O(n^2 \log n)$?
- $O(n^2)$

假设数字都不相同，那我们枚举中位数是什么，把比它大的数看作1，小的数看作-1，自己看成0。这样我们可以得到一个新的序列。

如果只考虑奇数长度的区间，显然这个枚举的这个数能成为中位数的区间数量就是包含0而且区间和等于0的区间数量。

这个是很好 $O(n)$ 统计出来的。

现在有相同的数字，也要考虑偶数长度的区间，只需略微修改即可。

关键结点

- 尝试人数：2
- 通过人数：1

- TAG：贪心，动态规划。

关键结点 solution 1

首先，对于给定的 m ， $h \leq \frac{n}{m}$ 。

我们选深度最深的结点 u ，设 v 是距离 u 为 h 的祖先。第一个关键结点设在 v ，然后就可以砍掉子树 v 。子树 v 上至少有 $h + 1$ 个结点。如此反复 m 次，我们至少可以砍掉 $(h + 1) \cdot m$ 个结点。

当 $h = \frac{n}{m}$ 时，就能砍掉所有结点。因此对于给定的 m ， $h \leq \frac{n}{m}$ 。

关键结点 solution 1

然后对于任意 k ，我们可以贪心地求出满足 $h(v_1, \dots, v_m) = k$ 的最小的 m 。方法是，设当前没有被覆盖的结点集合为 S 。从 S 中选出深度最深的结点 u ，设 v 是距离 u 为 k 的祖先。

此时如果要在子树 v 上增加一个关键结点，那么任何一种方案不会比把 v 设为关键结点更优。

这时候把 v 设为关键结点并把半径 k 以内的所有点标记。重复操作直到 S 为空集。由于 $m < \frac{n}{h}$ ，因此最坏复杂度是 $O(\frac{n^2}{h})$ 。树退化成链的情况下，复杂度为 $O(n)$ 。

关键结点 solution 2

对于给定 k ，求出满足 $h(v_1, \dots, v_m) = k$ 的最小的 m ，也可以使用 $O(n)$ 的类似DP，维护每个子树在贪心的情况下如果根不是关键节点未被覆盖点的最大深度。

对于随机的数据，方法一的效率远小于 $O(n^2 \log n)$ ，甚至快于方法二，时限也比较宽松，可以轻松AC。

填数字

- 尝试人数：6
- 通过人数：0

- TAG：并查集，计数。

填数字

设 $f(x)[i]$ 代表第 i 个位置填 $[1..x]$ 的合法方案数。

假设我们知道了 $f(x)[i]$ ，要算 $f(x+1)[i]$ ：

- 如果所有和 i 相关的限制中的 c 都大于 x ，则方案直接加一。
- 如果和 i 相关的限制中的 c 有等于 x 的。则合并该限制中的两个位置，方案统计为原来两个位置方案的相乘，并且删除限制。处理完这些限制后，再把新位置的方案加一。

如果没有限制的话，方案都是累加的，所以枚举 x 的时候可以只计算和限制有关的位置，最后再更新所有位置即可。如果碰到限制条件就用并查集来合并以及更新方案。

- 尝试人数：3
- 通过人数：0

- TAG：暴力搜索。

读懂题

只需枚举单位之间的攻击顺序，每个单位移动到的位置以及攻击哪个单位即可。

注意题目给出的种种要求与限制，范围非常小。

大部分trick都在样例里面，敢写敢过。

谢谢各位