

T1 膜拜大佬

- 题面有点绕。
- 每道题有个得分范围，以及所有题目的分数总和。豆豆选择尽可能多删掉大佬的某些题目的代码，然后告诉大佬说这样大佬也比他犇。大佬肯定想装弱，但由于豆豆比较了解大佬，所以大佬得分方案无论是什么样的（但每道题得分依旧在豆豆猜测的范围以内），都不会比豆豆低。

算法1：签到

- 豆豆爆零了，自然大佬可以啥都不做也不比豆豆低。

算法2：爆搜

- 留了不少的部分分给爆搜。

算法3：错误的贪心

- 最多去掉多少题，就等于最少做多少题。
- 那么直接做 L_i 最高的几个题。

算法4：正确的贪心

- 上面考虑掉了另一个情况，就是可以根据不做的题，最多可能又多少分，到推出剩下的题目最少又多少分。
- 那么把所有题目的最高分从小到大排序，选择丢掉这些最高分最小的 不值钱的题。
- 然后和上面的贪心取个最优就行。

T2 加密货币

- 题面紧跟时事。
- 题面反过来就是把原串分成两个子串，两个串互为翻转。（注意不是对于每一位的翻转，是整个串的翻转）

算法1：爆搜

- 长度不超过10，那就枚举一下哪些属于左边的串，哪些属于右边的串，然后再检查。

算法2：爆搜

- 长度不超过20，当然也能爆搜了，只是需要写的高效一些，剪枝去掉一点不合法的情况。

算法3：爆搜

- 长度不超过40，可以尝试如何折半爆搜。
- 假如我们知道了左边一半20个字符是如何分成两个串 (S_A, S_B) ，那么右边只能是 (S_B^T, S_A^T) 。
- 所以爆搜左边所有可能性，把 (S_A, S_B) 放进hash表。再爆搜右边的所有可能性，把 (S_B^T, S_A^T) 到hash表查询有多少能配上的。

T3 联谊活动

柠檬树上柠檬果，柠檬树下

- 这是一个构造题，需要想清楚怎么样构造最多的朋友，且又符合条件。
- 图论角度来说，就是一个二分图，每个点有至少 D 条边，最大匹配不超过 A 。假设二分图两边各有 N_1, N_2 个点, $N_1 \geq N_2$ 。

算法0：Cout << -1

- 每次输出一个数？无解输出 -1 ？这不直接先 `cout << -1;` 一个。
- 恭喜获得 15 分。

算法1：爆搜

- 遇事不决，先搜一个再说。只要题目读懂了，没有不能搜的，甚至还能打表。

算法2：特判

- $A = \min(N_1, N_2)$ ，这不是所有边都给连上？
- $A < 2 * D$ ，这肯定无解，否则一定有解。

算法3：讨论讨论

- 限制最大匹配的同时，还要尽可能的塞边，那么一定是 N_1 所有点连向 N_2 前 V 个点， N_2 连向 N_1 前 $A - V$ 个点，一共 $N_1 \times V + (N_2 - V) \times (A - V)$ 。
- 那么 V 是多少呢？会发现边数实际上是一个二次函数，所以我们试一下二次函数的边界值和极值就可以找到最大的了。
- 即边界 $V = 0, D, A - D, N_1, N_2$ 和极值 $V = \frac{A+N_2-N_1}{2}, \frac{A+N_2-N_1}{2} + 1$ 。

T4 开宝箱

算法1：爆搜

- 没啥好说的

算法2：爆搜+剪枝

- 可以过大一点的数据。

算法3： $O(N^4)$

- 可能能有同学发明这个复杂度吧

算法4：DP

- 先 S, T 做个差。然后我们要对很多个区间 $+1, -1$ 使得差为 0 。
- 注意到最优解中正负区间不会重叠。
- 那 $DP[i][sum]$ 表示当前清空了前 i 个位置，目前这些区间和为 sum 。显然 $sum \in [-10 * i, 10 * i]$ 。每次合法决策只有 $O(i)$ 种（因为算上当前的值要 $\%10=0$ ）。
- 复杂度 $O(N^3)$ 。

算法5：DP+剪枝

- 注意到从一个位置为起始的区间最多有 9 个（10个的话就可以消除一下）。
- 那么决策也就 $O(1)$ 了。
- 复杂度 $O(N^2)$ 。