

《基础 ABC 练习题》解题报告

浙江省衢州第二中学 方心童

2024 年 10 月 9 日

目录

1	题目描述	2
2	数据范围	2
3	评分细则	3
4	解题过程	4
4.1	角度一	4
4.1.1	算法 1.1	4
4.1.2	算法 1.2	4
4.1.3	算法 1.3	4
4.2	角度二	5
4.2.1	算法 2.1	5
4.2.2	算法 2.2	6
4.2.3	算法 2.3	6
4.2.4	算法 2.4	7
4.3	总结	7
5	参考资料	7

1 题目描述

给定两个集合 S_1, S_2 ，定义一个长度为 $3n$ 且仅包含 ABC 三种字符的串 s 是好的，当且仅当存在一种方案将 s 划分成 n 个长度为 3 的子序列，且这 n 个子序列都是 ABC，BCA 或 CAB。设 n 个子序列中 ABC 的个数为 x ，BCA 的个数为 y ，还要求 $x \in S_1, y \in S_2$ 。

现在有一个长度为 $3n$ 的字符串 s ，字符串中仅包含 ABC? 四种字符，你需要计算将所有 ? 都分别替换成 ABC 三个字符中的某一个的方案，使得串 s 是好的。

对 2^{32} 取模。 T 组测试数据。

2 数据范围

Subtask	分值	特殊限制
1	20	s 中没有 ?，且 $ S_1 = S_2 = n + 1$
2	20	s 中没有 ?
3	10	s 中只有 ?，且 $ S_1 = S_2 = n + 1$
4	20	$ S_1 = S_2 = n + 1$
5	30	无特殊限制

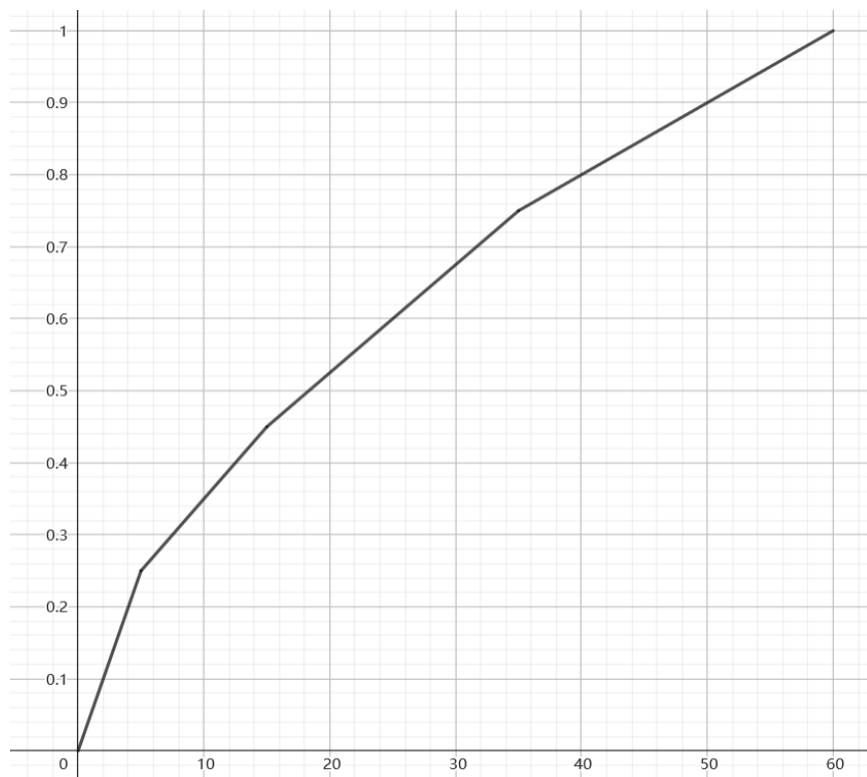
对于所有数据，保证 $T = 60$ 。对于每个测试点内的第 i 组测试数据，保证 $n = i$ 。

3 评分细则

你可以选择是否回答每组数据。对于每组数据，会根据你的输出给出你在这组数据上的评分系数 $p \in [0, 1]$ 。首先，你的程序需要正常结束并且所有你选择回答的答案均正确，否则 $p = 0$ 。在此基础上，记 d 为在所有数据中你的程序选择回答的最大的 n ，则有

$$p = \begin{cases} \frac{1}{20}d & d \leq 5 \\ \frac{1}{4} + \frac{1}{50}(d - 5) & 5 < d \leq 15 \\ \frac{9}{20} + \frac{3}{200}(d - 15) & 15 < d \leq 35 \\ \frac{3}{4} + \frac{1}{100}(d - 35) & 35 < d \leq 60 \end{cases}$$

p 与 d 的大致图像如下图所示。



4 解题过程

本题为 AGC055D 的加强版。接下来会以两个不同的角度来刻画这个问题。

4.1 角度一

4.1.1 算法 1.1

考虑对于一个已经确定的 s 怎么判定是否合法。

枚举 x, y 表示最终序列中 ABC 和 BCA 的个数, 记 $p_{o,i}$ 表示 s 序列中第 i 个 o 字符, 则会贪心地将 $p_{a,i}$ 分配给第 i 个 ABC, 对于 BCA, CAB 同理。在分配完第一个字符后, 仍可贪心地分配第二、三个字符, 也即将 $p_{b,y+i}, p_{c,n-x+i}$ 分配给第 i 个 ABC, 其余同理。

因此只需要枚举 x, y , 最优的分配方式是确定的, 于是可以 $O(n^3)$ 判定某个字符串 s 是否合法。可以通过子任务 1, 2, 获得 40 分。

4.1.2 算法 1.2

考虑给定 x, y 如何计数, 则上面的最优分配方案可以转化为对于 $p_{a,i}$ 和 $p_{b,y+i}$ 等的顺序限制, 然后设 $f_{a,b,c}$ 表示当前填了前 a 个 A, b 个 B, c 个 C 的方案数。这个 DP 是 $O(n^3)$ 的。

然后对于一般的情况, 只需要枚举哪些子集对是合法的, 容斥即可计算答案。注意到如果两个子集 \max 加起来大于 n 肯定是无解的, 因此只有 $O(2^n n)$ 个子集对需要计算, 时间复杂度 $O(2^n n^4)$, 可以通过 $n \leq 15$, 期望得分 45。

4.1.3 算法 1.3

算法 1.2 是指数级的, 我们尝试优化枚举集合的过程。考虑 $|S_1| = |S_2| = n + 1$ 的情况, 假设现在 ABC 个数的子集已经确定, 则前 i 个 A 已经确定了。在考虑要将哪些 B 分配进去的时候, 会对于 y 有一个限制, 可以表示

为 y 要大于某个值。对于剩下的分配，限制也总能表示成 y 要大于某个值或者 y 要小于某个值。换言之， y 的取值是一个区间。

在得到这个结论以后，只需要判定 y 取值区间非空即可。对于这样的判定，可以使用点减边容斥：枚举每个 $i \in [0, n]$ ，计算 $y = i$ 合法的方案数，然后对于每个 $i \in [1, n]$ ，减去 $y = i - 1, y = i$ 同时合法的方案数，即可得到 y 取值区间非空的方案数。对于不满足特殊性质的情况，只需要将所有 S_1, S_2 中的元素拿出来做这个容斥即可。

同理，对于 x 的枚举也可以应用这样的优化，因此总共只需要枚举 $O(n^2)$ 个集合对，时间复杂度 $O(n^5)$ ，常数较大，笔者的代码在互测 OJ 上用 2.7s 左右，期望得分 95 ~ 100。

4.2 角度二

4.2.1 算法 2.1

仍然考虑判定一个确定的 s ，不同的是加上特殊性质 $|S_1| = |S_2| = n + 1$ 考虑。我们有如下引理：

引理 1. 记 $sum_{o,i}$ 为 s 序列前 i 个数 o 字符的出现次数，记

$$A = \max_{1 \leq i \leq 3n} sum_{a,i} - sum_{c,i}$$

$$B = \max_{1 \leq i \leq 3n} sum_{b,i} - sum_{a,i}$$

$$C = \max_{1 \leq i \leq 3n} sum_{c,i} - sum_{b,i}$$

则序列合法充要于 $A + B + C \leq n$ 。

证明. 首先证明必要性。在 ABC、BCA、CAB 中，只有 ABC 中 A 出现在 C 前面，因此， $A \leq$ ABC 的出现次数。对于 B, C 同理。由于 ABC、BCA、CAB 总出现次数为 n ，可以得到 $A + B + C \leq n$ 。

然后证明充分性。考虑将第 i 个 A，第 $i + B$ 个 B，第 $i + B + C$ 个 C 组合在一起（如果超过 n 视作循环）。如果 $i + B$ 没有循环，根据 A, B, C

的定义, 第 i 个 A, 第 $i+B$ 个 B, 第 $i+B+C$ 个 C, 第 $i+A+B+C$ 个 A 在序列上依次排布。又因为 $A+B+C \leq n$, 所以这些字符在同一个周期内, 放到原序列上就是轮换的。□

因此我们得到一个 $O(n)$ 判定某个字符串是否合法的方法, 期望得分 20。

4.2.2 算法 2.2

考虑将上述引理扩展到更一般的情况。具体的, 我们有如下引理:

引理 2. A, B, C 定义同引理 1 且满足 $A+B+C \leq n$ 。对于每个三元组 (x, y, z) 满足 $x+y+z=n, x \geq A, y \geq B, z \geq C$, 都满足存在一种对 s 的划分使得恰好划分成 x 个 ABC, y 个 BCA, z 个 CAB。

证明. 考虑对 n 归纳。假设当前 s 长度为 $3n$ 并且满足 $A+B+C \leq n$ 。对于 $A+B+C=n$ 的情况由引理 1 已经成立。对于 $A+B+C < n$, 选取第一个 A, 这个 A 后第一个 B, 这个 B 后第一个 C 组成一个子序列。如果选不出来会得到 $A+B+C \geq n$, 这是矛盾的。在将这个子序列删除之后, B, C 均不会改变, A 至多减少 1。对于选择子序列 BCA 以及 CAB 是同理的, 归纳成立。□

通过这个结论就可以 $O(n)$ 计算更一般的情况了, 期望得分 40。

实际上, 引理 2 与角度一中“取值是一段区间”的结论是对应的。

4.2.3 算法 2.3

考虑进行 DP, 设 $f_{a,b,c,A,B,C}$ 表示填了前 a 个 A, b 个 B, c 个 C, 最大值为 A, B, C 的方案数。时间复杂度 $O(n^6)$, 常数极小, 可以通过 $n \leq 25$, 期望得分 60。

4.2.4 算法 2.4

只需要在算法 2.3 的基础上对这个 DP 进行优化。枚举最终 A, B 的值是什么, 则可以求出 C 的上界。记 $f_{a,b,c,0/1,0/1}$ 表示填了前 a 个 A, b 个 B, c 个 C, 且前两个上界是否达到的方案数, 时间复杂度降至 $O(n^5)$, 常数不大, 笔者的代码在互测 OJ 耗时约 1.3s, 期望得分 100。

4.3 总结

角度一是笔者最开始的做法, 并不需要给出一个精确的刻画, 通过比较多套路性的观察解决了这个问题。角度二中引理 1 是原题题解给出的做法, 比较精确地刻画了合法的充要条件, 引理 2 是笔者与集训队成员刘海峰讨论后得到的扩展, 并获得了更小的常数。两个角度殊途同归, 并且某些结论存在比较强的关联, 最终的代码实现也比较类似。

5 参考资料

AGC055D

AGC055D 官方题解

本题在命制过程中有过与刘海峰、郑钧、陈昕阳的交流。