

# 《卡牌游戏》解题报告

## 题目大意

给定一个长度为  $n$  的序列  $A_1 \dots A_n$ 。保持相对顺序不变拿出  $A$  一个子序列  $B_1 \dots B_m$ ，最大化：

$$\sum_{i=1}^{m-1} \lfloor \frac{S}{\text{LCM}(B_i, B_{i+1})} \rfloor \times \text{LCM}(B_i, B_{i+1})$$

另外，你有一个参数  $tp$ 。若  $tp = 1$ ，你还需要对于  $i = 1 \dots n$ ，求出  $A_i$  不被拿出的情况下上式最大是多少。

## 数据范围与约定

| 子任务编号 | 分值 | $n \leq$        | $S \leq$        | 特殊性质                   |
|-------|----|-----------------|-----------------|------------------------|
| 1     | 5  | $10^3$          | $10^5$          | 无                      |
| 2     | 5  | $10^4$          | $10^5$          | 无                      |
| 3     | 10 | $10^5$          | $3 \times 10^5$ | $A_i$ 在 $[1, S]$ 中均匀随机 |
| 4     | 10 | $10^5$          | $3 \times 10^5$ | $A_i \leq 100$         |
| 5     | 10 | $10^5$          | $10^5$          | $tp = 0$               |
| 6     | 10 | $10^5$          | $10^5$          | 无                      |
| 7     | 10 | $3 \times 10^5$ | $3 \times 10^5$ | $tp = 0$               |
| 8     | 10 | $3 \times 10^5$ | $3 \times 10^5$ | 无                      |
| 9     | 15 | $5 \times 10^5$ | $5 \times 10^5$ | $tp = 0$               |
| 10    | 15 | $5 \times 10^5$ | $5 \times 10^5$ | 无                      |

时间限制：3s

空间限制：512MB

## 约定

记  $tr(x, y) = \lfloor \frac{S}{\text{LCM}(x, y)} \rfloor \times \text{LCM}(x, y)$ 。

## 算法一

我会暴力dp！

设  $f_i$  表示考虑完前  $i$  张卡牌，钦定取第  $i$  张的最大价值。转移为  $f_i = \max_{j=1}^i (f_j + tr(a_j, a_i))$ 。

第二问容易用前后缀的 dp 值拼起来。

时间复杂度  $O(n^2 \log S)$ ，期望得分 5。

## 算法二

我会快速计算 gcd!

使用  $O(S) - O(1)$  或者更简单的做法计算  $tr(x, y)$ 。

复杂度  $O(S + n^2)$ 。期望得分 10。

## 算法三

我会均匀随机!

容易发现:

$$tr(x, y) = \max_{z \leq S, x, y | z} z$$

在 dp 的过程中, 维护辅助数组  $g_i = \max_{a_j | i} f_j$ , 转移可以写为  $f_i = \max_{a_i | j} (g_j + j)$ 。

当  $a_i$  在  $[1, S]$  内均匀随机时,  $\frac{S}{a_i}$  期望为  $\ln S$ 。

复杂度  $O(n \ln S)$ 。结合算法二, 期望得分 20。

## 算法四

我会  $a_i$  很小!

可以直接枚举上一次转移的  $a$  值是多少。

复杂度  $O(100n)$ 。结合算法三, 期望得分 30。

事实上我们还有更强的结论: 注意到  $tr(x, y) = 0$  或  $tr(x, y) > \frac{S}{2}$ 。若  $S \geq (\max a)^2$ , 最优策略一定是将能选的都选上。因为考虑到此时任意  $tr(a_i, a_j) > \frac{S}{2}$ , 如果存在不选一定不优。虽然看起来没啥用, 但出题人并没有造  $S < (\max a)^2$  的数据。

## 算法五

我会低于平方!

以下假设  $n, S$  同阶。

考虑对时间分块, 以此加速算法一中的 dp。

块内的转移暴力。块间的转移在块与块之间的分界点处被预处理。

每当一个块结束后, 我们会暴力重构算法三的  $g$  数组, 这样跨越这个分界点的转移都可以被快速考虑。

复杂度  $O(n\sqrt{n \log n})$ 。期望得分 50。

我知道它看起来很萌但不失为一种做法。另外, 我的一位同学声称他会一个  $O(n^{\frac{5}{3}})$  的算法, 但我不是很懂。

## 算法六

我会根号分治!

考虑一次转移  $a_j \rightarrow a_i$ , 若  $a_j \leq \sqrt{S}$  或  $a_i \leq \sqrt{S}$ , 可以直接枚举不超过根号一边的值; 若  $a_i > \sqrt{S}$  且  $a_j > \sqrt{S}$ , 可以用算法三。

复杂度  $O(n\sqrt{n})$ 。常数比较大, 且如果没有发掘进一步的性质比较难做  $tp = 1$ , 期望得分 70。

## 算法七

我会卡常！

首先，对算法四中的结论进行扩展，可以得到，若  $f_i = f_j + tr(a_j, a_i)$ ，一定满足：  $[j, i]$  中  $\leq \sqrt{S}$  的数不超过 3 个。因此，对于两边存在至少一个  $\leq \sqrt{S}$  的转移，就可以直接做到线性了。

其次，若  $f_i = f_j + tr(a_j, a_i)$ ，必有  $f_j + S > f_{i-1}$ 。我们可以维护一个指针，指向可以转移的第一个位置。如果可以转移的位置很少就暴力做。将这个做法和算法三平衡一下后，出题人认为其常数是优秀的。想将这个东西卡满的一种方式是取大于  $\sqrt{S}$  的最小的  $\sqrt{S}$  个质数。但显然常数比算法六快了将近 4 倍。

另外，怎么做  $tp = 1$  呢？还是考虑前后缀拼接的方法，可能的拼接次数有了类似结论后也是  $O(n\sqrt{n})$  级别的。现在问题变为快速区间取  $\max$ ，查询最后每个位置的值，可以直接猫树。

时间复杂度  $O(n\sqrt{n})$ ，空间  $O(n \log n)$ 。常数较小。std 跑了 1.4s，时限超过其两倍。期望得分 100。

鉴于出题人比较菜，如果被爆标了请轻喷。