CSP-S 2020 模拟赛

第二试 题目解析

时间：2020 年 10 月 dd 日 hh:mm ~ hh:mm

AHSOFNU
PinkRabbit

题目名称	朝比奈实玖瑠的采购	长门有希的序列	古泉一树的游戏
题目类型	传统型	传统型	传统型
输入文件名	asahina.in	nagato.in	koizumi.in
输出文件名	asahina.out	nagato.out	koizumi.out
每个测试点时限	1.0 秒	3.0 秒	1.0 秒
内存限制	512 MB	512 MB	512 MB
测试点数目	25	10	25
测试点是否等分	是	是	是

提交源程序文件名

对于 C++ 语言	asahina.cpp	nagato.cpp	koizumi.cpp

朝比奈实玖瑠的采购（asahina）

【题目大意】

有 n 个盒装的茶叶，你可以购买其中的一部分，每一盒只能买一次。

对于第 i 个盒装的茶叶，其中含有 c_i 包茶叶，每包茶叶的清香度为 w_i，这一盒的价格为 v_i。

有 m 种茶水的制作方法，这些方法也不能重复使用。

第 j 种方法需要 C_j 包茶叶，每包茶叶的清香度都应至少为 W_j，制作后可以增加 V_j 的收入。

你可以购买 n 盒茶叶其中的一部分，然后选择一些制作方法来获得收入。

请你求出利润的最大值，也即最大化**卖茶水带来的收入**与**购买茶叶的支出**的差。

【数据范围】

对于所有测试点：$1 \le n, m \le 2000$，$1 \le c_i, C_j \le 50$，$1 \le w_i, W_j, v_i, V_j \le 10^9$。

每个测试点的具体限制见下表：

<table>
<tr><th>测试点编号</th><th>$n \le$</th><th>$m \le$</th><th>特殊限制</th></tr>
<tr><td>$1 \sim 4$</td><td>15</td><td>2000</td><td rowspan="2">无</td></tr>
<tr><td>$5 \sim 8$</td><td>2000</td><td>15</td></tr>
<tr><td>$9 \sim 12$</td><td colspan="2">100</td><td>$c_i = C_j = 1$</td></tr>
<tr><td>$13 \sim 16$</td><td colspan="2" rowspan="3">2000</td><td>$w_i = W_j = 1$</td></tr>
<tr><td>$17 \sim 20$</td><td>$v_i = V_j = 1$</td></tr>
<tr><td>$21 \sim 25$</td><td>无</td></tr>
</table>

【算法一（100 分）】

对每盒茶叶以及每种方法按照清香度从大到小排序。

按顺序考虑每种方法，能满足这种方法的清香度需求的茶叶是一个前缀。

结合另外两维：茶包数与价值，这启发我们考虑背包问题，并使用动态规划解决。

令 $\mathrm{f}[i][j]$ 表示考虑了清香度 $\ge i$ 的所有茶叶和方法后，此时剩余还未使用的茶包的个数为 j 时，能够确定的利润的最大值（可能为负）。

对于茶叶：从 $\mathrm{f}[*][j]$ 转移到 $\mathrm{f}[*][j+c]$，代价为 $-v$。

对于方法：从 $\mathrm{f}[*][j]$ 转移到 $\mathrm{f}[*][j-C]$，代价为 V。

任何时刻 $\mathrm{f}[i][j]$ 中的 j 值均不能小于 0（不能透支清香度高的茶叶）。

为了节省空间，可以把 i 所在的那一维滚动掉。

时间复杂度为 $\mathcal{O}((n+m)\sum c)$，期望得分 100 分。

长门有希的序列（nagato）

【题目大意】

请你维护一个长度为 n 的整数序列 $[a_1, a_2, \ldots, a_n]$，执行 q 次操作，有如下两种操作：

1. `1 l r`：将区间 $[l, r]$ 中的每个元素变为自身的平方。即对每个 $l \le i \le r$，执行 $a_i \leftarrow a_i^2$。
2. `2 l r`：询问区间 $[l, r]$ 中的所有元素的和，对 $p = 998244353$ 取模。即输出 $\left(\sum_{i=l}^{r} a_i\right) \bmod p$。

【数据范围】

对于所有测试点：$1 \le n, q \le 2 \times 10^5$，$1 \le a_i < p$，$\mathrm{op} \in \{1, 2\}$，$1 \le l \le r \le n$。

其中 $p = 998244353$。

每个测试点的具体限制见下表：

测试点编号	$n, q \le$	特殊限制
1	5000	无
$2 \sim 3$	2×10^5	对于 1 操作有 $l = r$
$4 \sim 5$		对于 2 操作有 $l = r$
$6 \sim 10$		无

【算法一（10 分）】

暴力模拟两种操作。

时间复杂度为 $\mathcal{O}(qn)$，期望得分 10 分。

【算法二（20 分）】

对于 1 操作有 $l = r$ 时，相当于单点修改区间求和。

使用线段树或树状数组维护即可。

时间复杂度为 $\mathcal{O}(n + q \log n)$，期望得分 20 分。

【算法三（100 分）】

假设 a 序列的某个位置的初始值为 v。考虑对 v 进行若干次 1 操作后的结果：

形成序列 $[v^1, v^2, v^4, v^8, v^{16}, v^{32}, \ldots]$。形式化地，每一项有通式 v^{2^i}。

注意到 $p = 998244353$ 是质数，应用费马小定理：$v^{2^i} \equiv v^{2^i \bmod (p-1)} \pmod p$。

注意到 $p - 1 = 119 \cdot 2^{23}$，所以序列 $e_i = 2^i$ 在模 $p - 1$ 意义下会在 23 步内进入循环节。

而循环节长度，即 $\mathrm{ord}_{119}(2)$，应为 $\varphi(119) = \varphi(7 \cdot 17) = 96$ 的因数，实际计算可知长度为 24。

回到原问题，考虑使用线段树维护，在节点上记录子树中的所有位置是否都已经进入循环节。

如果没有进入，修改时递归到底层；否则打标记，只需处理长度为 24 的数组的循环移位。

时间复杂度为 $\mathcal{O}(24(n + q) \log n)$，期望得分 100 分。

古泉一树的游戏（koizumi）

【题目大意】

试求三行 Chomp 游戏的策略。

规定序列 $[a_1, a_2, \ldots, a_n]$ 对应了 n 行 Chomp 游戏的一个状态，具体地说，对应了第 i 行的长度为 a_i 的状态。

每个测试点有 T 组数据。

【数据范围】

对于所有测试点：

$1 \le T \le 1000$，$1 \le n \le 3$

$1 \le a_1, a_2 \le 10^9$（如果 a_2 存在）

$1 \le a_3 \le 1000$（如果 a_3 存在）

保证输入的是一个合法的状态，且至少有一种状态转移方式，即 $a_i \ge a_{i+1}$ 且 $a_1 \ge 1$。

每个测试点的具体限制见下表：

测试点编号	$n \le$	特殊限制
1	1	无
2	2	$a_1 \le 9$
3		$a_1 \le 10^3$
4		无
$5 \sim 7$	3	$a_3 = 1$
$8 \sim 10$		$a_3 \le 3$
$11 \sim 13$		$a_1 \le 80$
$14 \sim 18$		$a_3 \le 80$
$19 \sim 23$		$a_3 \le 150$
$24 \sim 25$		无

【算法一（4 分）】

对于 $n = 1$ 的情况，当 $a_1 = 1$ 时输出 `Lose`，否则输出 `1`。

期望得分 4 分。

【算法二（16 分）】

对于 $n = 2$ 的情况（$n = 1$ 时算作 $a_2 = 0$ 的 $n = 2$ 的情况）：

不难发现 $[a_1, a_2]$ 是必败态当且仅当 $a_2 = a_1 - 1$。

所以如果 $a_2 = a_1 - 1$ 输出 `Lose`，否则如果 $a_2 = a_1$ 输出 $[a_1, a_1 - 1]$，否则输出 $[a_2 + 1, a_2]$。

期望得分 16 分。

【算法三（28 分）】

对于 $0 \le a_3 \le 1$ 的情况（$n = 2$ 时算作 $a_3 = 0$ 的情况）：
当 $a_3 = 0$ 时已经在算法二中考虑过了，所以仅考虑 $a_3 = 1$。
不难证明仅有 $[3, 1, 1]$ 和 $[2, 2, 1]$ 是当 $a_3 = 1$ 时的必败态。
如果输入的是必胜态，要么可以转移到 $[3, 1, 1]$，要么可以转移到 $[2, 2, 1]$。
除了 $[1, 1, 1]$ 和 $[2, 1, 1]$，它们分别转移到 $[1, 0, 0]$ 和 $[2, 1, 0]$。
期望得分 28 分。

【算法四（40 分）】

在算法三的基础上，考虑 $2 \le a_3 \le 3$ 的情况。
对于 $a_3 = 2$，可以证明 $[a_1, a_2, 2]$ 是必败态当且仅当 $a_2 = a_1 - 2$。
对于 $a_3 = 3$，可以证明仅有的三个必败态是 $[6, 3, 3]$、$[7, 4, 3]$ 和 $[5, 5, 3]$。
如果输入的是必胜态，有更多的细节需要处理，请读者自行考虑。
期望得分 40 分。

【算法五（12 分）】

对于 $n = 3$ 且 $a_1 \le 80$ 的情况，可以设计一个三维的动态规划状态：
令 $\mathrm{f}[a][b][c]$ 表示第一行有 a 列，第二行有 b 列，第三行有 c 列的游戏是否是必胜态。
状态转移时枚举要选取哪个格子，有 $\mathcal{O}(a + b + c)$ 种转移方式。
预处理的时间复杂度为 $\mathcal{O}\Big((\max a_1)^4\Big)$。
记录下范围内的所有必败态，在回答询问时，直接枚举这些必败态进行判断。
当 $a_1 \le k$ 时，猜想必败态数量是 $\mathcal{O}(k^2)$ 的。
总时间复杂度为 $\mathcal{O}\Big((\max a_1)^4 + T(\max a_1)^2\Big)$。
期望得分 12 分，结合算法四期望得分 52 分。

【算法六（100 分）】

请阅读同目录下的 ***chomp/study-3-rowed-chomp.pdf***。
使用其中图解计算过程中提到的方法，用程序实现后表现优异。
时间复杂度为 $\mathcal{O}(k^3 + Tk)$，其中 $k = \max a_3$。
这里的时间复杂度记号基于几个文中提到的猜想，它们在本题的数据范围内可以看作是成立的。
由于算法的固有特征，程序的时间常数较小，所以可以在规定的时限内通过本题。
代码中使用了并查集等优化，但效果并不明显，朴素实现就可以做到几乎相同的常数。
期望得分 100 分。

【算法七（72 ∼ 92 分）】

如果你实现了在同目录下的 ***chomp/study-3-rowed-chomp.pdf*** 中提到的比较劣的打表程序。
也就是直接从算法五继承而来，再加上寻找循环节的部分。其时间复杂度应该是 $\mathcal{O}(k^4)$ 的。
应该能在规定时间内通过 $a_3 \le 80$ 的测试点。

对于 $a_3 \leq 150$ 的测试点，得到了打出的表后，可以尝试将表压缩（或不用压缩，表的大小并不算太大，是否要压缩取决于代码长度限制），然后硬编码进程序中，不需要预处理，询问时直接访问状态即可。

期望得分 $72 \sim 92$ 分。