NOI 模拟赛 Day2 题解

2025 年 4 月 30 日

目录

1 构造题

1.1 算法一

下文中令 $m = 10^K$。

构造一条链出来，$X = T = k$，期望得分 5 分。

在这个基础上进行一些神秘的乱搞可能可以通过 Subtask2，期望得分 15 分。

1.2 算法二

考虑利用二进制来构造使得 $f_{i,i} = 2^i$，如下图所示：

1	1		
1	2	2	
	2	4	4
		4	8

显然这样可以做到 $X = 3\log m, Y = \log m$，可以通过 Subtask3，可能可以通过一些乱搞通过 Subtask4，期望得分 $25 \sim 35$ 分。

1.3 算法三

考虑更换其他进制进行尝试，发现换成三进制或六进制可以有效减少 Y，但会导致 X 变大。

1	1	1		
1	2	3		
1	3	6	6	6
		6	12	18
		6	18	36

1	1	1		
1	2	3	3	
1	3	6	9	9
	3	9	18	27
		9	27	54

六进制中 $X = 8\log_6 m, Y = 2\log_6 m$，可以通过 Subtask 5。

三进制中 $X = 5\log_3 m, Y = \log_3 m$，可以通过 Subtask 6。

期望得分 $35 \sim 55$ 分。

1.4 算法四

猜测本题的做法肯定是找到一个合适的进制，用该进制能做到将 X, Y 都足够小，通过打表可以得到在 $5, 8, 13, 21, 34 \ldots$ 进制最优，于是猜想最终图片状态和斐波那契数列有关。也可以通过观察性质并猜测结论注意到以下棋盘：

1	1	1		
1	2	3	3	
	2	5	8	8
		5	13	21
			13	34
				34

注意到该棋盘与斐波那契数列有关，众所周知，斐波那契数列的第 n 项是 $O(\phi^n)$ 级别的，其中 $\phi = \frac{1+\sqrt{5}}{2}$。此时显然 $X = 2\log_\phi m$，符合要求。

考虑最坏情况下 Y 的级别，如果一组数据存在 f_i, f_{i+1} 同时被选择但 f_{i+2} 没有被选择，那么可以将 f_i, f_{i+1} 替换为 f_{i+2}。以此归纳可以得到不会出现选择斐波那契数列中相邻两个数。所以 $Y \le \frac{\log_\phi m}{2}$，计算得到 Y 也符合要求。期望得分 100 分。

出题人并不太会证明通过斐波那契数列构造是最优的，严谨证明了这个是最优情况的同学可以上台分享一下或者课下联系出题人。

2 匹配题

2.1 算法一

考虑最暴力的状压 dp，设 $dp_{i,j,S}$ 表示考虑到 i，删了 j 个位置，前面还有 S 集合没有被匹配的最大答案。

转移的时候考虑 i 是和前面 S 内的匹配还是加入 S 和后面的匹配，这两种情况都可以把绝对值函数拆贡献来计入答案。可以通过 $n = 30$ 的数据，期望得分 25 分。

2.2 算法二

可以尝试对算法一进行一些剪枝和随机化操作，也可以直接模拟退火或爬山，正确性不知道，复杂度不知道，可能可以通过 n 大一点的或者数据随机的数据，期望得分 $0 \sim 60$ 分。

2.3 算法三

对于 $S = \{a, b\}$ 的情况，注意到同一时刻中 S 集合内一定只有一种颜色。如果同时存在两种颜色那么一定不如内部解决一些更优。

所以可以直接记录 $dp_{i,j,k,0/1}$ 表示前 i 个位置，删了 j 个位置，剩余 k 个 a/b 没有匹配的最优答案，转移显然。复杂度 $O(Tn^2k)$，结合算法一期望得分 40 分。

2.4 算法四

考虑进一步扩展算法三的结论，在任意时刻，S 内的元素要么都是同一种颜色，要么它们都会和同一种颜色进行匹配，否则可以通过调整变得不劣。

记录 $dp_{i,j,k,c,0/1}$ 表示前 i 个位置，删了 j 个位置，S 内有 k 个元素，这 k 个元素都是 c 还是都与 c 匹配的最优答案。

转移就枚举 $i+1$ 的几种选择进行转移即可，这部分是平凡的，复杂度 $O(Tn^2k|S|^2)$，期望得分 100 分。

3 作业题

3.1 算法一

暴力建图或线段树优化建图网络流，期望得分 $8 \sim 16$ 分。

3.2 算法二

考虑贪心匹配。从左往右扫描每一份作业，每次将能覆盖当前作业的区间加入一个集合，当前作业显然贪心的匹配 r 更小的区间是更优的。用优先队列维护集合，复杂度 $O(n(n+m)\log m)$，期望得分 28 分。

3.3 算法三

对于 A 性质的情况，对于每个 i 可以选择的人是一段区间。这时一定每次贪心选择最左边的区间是更优的。考虑 Hall 定理，记录 $s_i = \{\sum_{l_j \le p \le r_j \le i} c_j - \sum_{k=p}^{i} a_k\}$，答案就是 $\sum_{l_i \le p \le r_i} c_i - \max_{i \ge p} s_i$。

扫描线 p，用线段树维护后者即可，只需要进行区间加，区间查询 max 操作。复杂度 $O((n+m)\log n)$，结合算法三期望得分 48 分。

3.4 算法四

对于 B 性质的情况，所有区间构成一棵树结构。每次需要考虑的区间是树上一条到根的链，而一定贪心的由内到外选择是最优的。类似算法三用 Hall 定理转化后树剖，用线段树维护即可。结合算法二，算法三期望得分 64 分。

3.5 算法五

我们发现完全没有性质的若干杂乱区间是很难做的，考虑用一些方法把一般情况转化到算法四或算法五的问题。

注意到这时关注的区间也不是毫无性质，它们都覆盖同一个点。考虑利用这个性质进行转化。我们从右向左扫描线，考虑每个区间对应的题目分为全部放在右边和有一部分放在左边两种情况。

我们肯定是希望所有的区间都能全部放在右边的，如果不能也要让尽可能少的题目扔给左边。容易发现扔给左边的题目数量是固定的。考虑这些题目我们希望扔到左边的哪些位置。根据上面的贪心算法，我们是希望尽量往更左边扔题目，因为这样我们就有更多的时间和机会去尝试解决这些题目。

我们记录右边 s_i 的前缀最大值位置为 $i_1 \sim i_k$，找到 $p \sim i_k$ 中左端点最靠左的左端点位置 x，那么我们会找到 x 对应区间的右端点之前的第一个 i_t，将 $s_{i_k} - s_{i_t}$ 这部分题目全部扔到 x 这个区间上，这样可以贪心的获得更多的时间来解决这些必须丢到前面的题目。

我们暴力的进行这个过程，直到右半部分不需要扔题目到左边为止，可以证明，操作次数为 $O(n\log n)$ 的。

记录一个状态的势能为右儿子最大值严格大于左儿子最大值的线段树节点数量。容易发现，每次操作会使得势能减少 1，$lca(i_t, i_k)$ 对应节点的势能会从 1 变成 0。而每次区间修改至多会带来 $O(\log n)$ 的势能，所以总的操作次数不超过 $O(n\log n)$。最终复杂度 $O(n\log^2 n + m\log n)$，期望得分 100 分。