

《雷同》命题报告

李劲逸

2023 年 10 月 19 日

目录

1	题目描述	2
1.1	题目大意	2
1.2	数据范围	2
1.3	子任务	2
2	解题思路	2
2.1	正解	2
2.1.1	问题的刻画	2
2.1.2	固定 d 序列后最小化代价	3
2.1.3	设计 dp	3
2.2	子任务算法	3
2.3	留下的问题	4
3	命题总结	4
3.1	相关题目	4
3.1.1	[NOI2023] 合并书本	4
3.1.2	[NOIP2004 提高组] 合并果子	4
3.2	总结	4
3.3	致谢	4
3.4	参考资料	4
	参考文献	4

1 题目描述

1.1 题目大意

有 n 本书，每本书初始重量值为给定的 w_i ，磨损值 v_i 为 0。

每次可以花费 $w_i + w_j + \min(v_i, v_j)$ 的代价将两本书合并为一本，合并形成的书重量值为 $w_i + w_j$ ，磨损值为 $2 \max(v_i, v_j) + 1$ 。

求将所有书合并成一本的最小代价。

1.2 数据范围

多测， $\sum n^2 \leq 10^8; 1 \leq w_i \leq 10^9$ 。

1.3 子任务

1. (5 分) $n \leq 6, T \leq 5$ 。
2. (15 分) $n \leq 12, T \leq 5$ 。
3. (15 分) $n \leq 30, T \leq 5$ 。
4. (15 分) $n \leq 80, T \leq 5$ 。
5. (20 分) $\sum n^3 \leq 5 \times 10^7$ 。
6. (5 分) 同一组数据内的所有 w_i 相同。
7. (25 分) 无特殊限制。

2 解题思路

2.1 正解

2.1.1 问题的刻画

考虑把合并过程建成一棵二叉树：合并书 u, v 时，新建一个节点 x 表示合并后的书，并让 u, v 作为 x 的儿子。

现在尝试计算一棵二叉树对应的总代价。

首先考虑重量值部分，记叶子节点的深度从小到大依次是 $d_1, d_2, \dots, d_n$ (根的深度为 0)，显然 d_i 表示书一共参与了多少次合并，显然要让重量值小的书对应深度大的叶子，换言之，将重量值 w 从大到小排序后，重量值部分的总代价是 $\sum_{i=1}^n w_i d_i$ 。

再考虑磨损值部分的贡献，可以按照如下方式描述：对树进行长链剖分后，记根所在链（即最长的长链）以外的所有长链所在集合为 S ，则磨损值部分的贡献为 $\sum_{path \in S} 2^{len(path)} - 1$ （其中 len 表示链上的点数）。因为 $|S|$ 为固定值 $n - 1$ ，故后文按照 $\sum_{path \in S} 2^{len(path)}$ 算。

2.1.2 固定 d 序列后最小化代价

接下来假设已经固定 $\{d_1, d_2 \cdots d_n\}$ ，怎么建树才能最小化磨损值部分的贡献。

我们按照如下方式描述建树过程：

一开始有 n 个子树，这 n 个子树都只有 1 个点，每个子树有一个树高属性 h 和一个“最终深度”属性，“最终深度”初始值设为 d_i ，每一轮我们要将“最终深度”是当前最大值 t 的子树两两配对，合并成“最终深度”为 $t-1$ 的。配对子树 u, v 产生 $2^{\min(h_u, h_v)}$ 的代价。现在的问题是怎样匹配能最小化总代价。

注意到我们匹配干的事情是让两棵树中高度较低的那个作为“轻儿子”，对应长链的长度不再增长，而较高的那个树继续增长。我们显然希望在这一轮匹配中把大的 h 截停了。

所以最优策略是：将要匹配的子树按树高 h 排序后，将相邻两个匹配（即第 $2k-1$ 大匹配第 $2k$ 大），因为这样截停了第 $2, 4, 6 \cdots$ 大的 h ，显然是最优的策略。

接下来化简计算总磨损值代价的计算过程。对于一个 t ，若深为 t 的叶子有 a 个，合并过程中在“最终深度”为 $t+1$ 的轮次，有 $2b$ 个“最终深度”为 $t+1$ 的，则这些叶子所在长链分别长 $ctz(b), ctz(b+1), \dots, ctz(b+a-1)$ ，贡献 $lowbit(b), lowbit(b+1), \dots, lowbit(b+a-1)$ 。（特别的，因为整棵树最长的链不产生贡献，所以令 $lowbit(0) = 0$ ）。

2.1.3 设计 dp

尝试用 dp 解决原问题。从根开始往下一层一层确定二叉树的形态，我们只关心当前层的节点数和之前的层一共安排的叶子数。

$f_{i,j}$ 表示当前层还剩 j 个叶子节点没有确定是不是叶子，之前所有层一共安排了 i 个叶子。

则转移有两种：

1. 结束当前层的分配，也就是钦定剩下的 j 个本层节点不是叶子，直接进入下一层。即 $f_{i,j} + (\sum_{k=i+1}^n w_k)$ 更新 $f_{i,2j}$ 。

2. 钦定一个本层节点是叶子。即 $f_{i,j} + [j \neq 1]lowbit(j-1)$ 更新 $f_{i+1,j-1}$ 。

初始值： $f_{0,1} = 0$ ，答案： $f_{n,0} - (n-1)$ 。

时间复杂度 $O(n^2)$ ，空间复杂度容易做到 $O(n)$ ，足以通过本题。

2.2 子任务算法

子任务 1, 2 是平凡的，可以使用搜索或状压 dp 解决。

子任务 3 留给更优秀的搜索、调整、基于指数状态数的 dp 剪掉不优决策（类似 [NOI2023] 合并书本正解）等做法。如果实现极为优秀，可能可以通过子任务 4。

子任务 6 引导选手进行简单的性质分析和建模，如果使用二叉树建模，则很容易发现完全二叉树对应该特殊情况下的最优策略。

子任务 5 对应完成性质分析，但 dp 设计不够优秀得到的 $O(n^3)$ 、 $O(n^2 \log n)$ 做法。

2.3 留下的问题

出题人能力有限，无法得到更优秀的做法，但也无法证明 $O(n^2)$ 是本题的最优复杂度。对本题感兴趣的同学可以进行更深入的思考。

3 命题总结

3.1 相关题目

下面两道经典题与本题有较强关联，其做法对本题的思考可能有启发作用。

3.1.1 [NOI2023] 合并书本

本题来源于出题人对 [NOI2023] 合并书本的题目改编。两道题部分的做法较为相似，但正解差距很大。

3.1.2 [NOIP2004 提高组] 合并果子

这是一道经典的贪心题，本题的二叉树建模被用于证明“合并果子”的贪心结论。

3.2 总结

总的来说，本题是一道偏向结论分析的 dp 题，思维难度中等，代码难度简单。

本题的结论并不容易直接猜出，需要选手循序渐进地进行推导，主要考察了选手刻画问题和设计 dp 的能力，思考本题有助于选手建立解决最优化问题的思维体系。

预计有多于一半的集训队选手可以通过本题。

3.3 致谢

感谢国家集训队教练提供的互测平台。

感谢崔隽章、钟自厚同学复核本命题报告，感谢刘昀恒同学检查算法的正确性。

3.4 参考资料

[NOI2023] 合并书本及相关题解。

与刘昀恒、崔隽章、钟自厚三位同学的讨论。