



从圆桌问题谈数据结构的综合运用

例. 圆桌问题(AH'99)

题目：圆桌上围坐着 $2n$ 个人。其中 n 个人是好人，另外 n 个人是坏人。如果从第一个人开始数数，数到第 m 个人，则立即处死该人；然后从被处死的人之后开始数数，再将数到的第 m 个人处死.....依此方法不断处死围坐在圆桌上的人。试问预先应如何安排这些好人与坏人的座位，能使得在处死 n 个人之后，圆桌上围坐的剩余的 n 个人全是好人。

输入：文件中的每一行都有两个数，依次为 n 和 m ，表示一个问题的描述信息。约束条件： $n \leq 32767$ ， $m \leq 32767$ 。

输出：依次输出每一个问题的解。每一个问题的解可以用连续的若干行字符表示，每行字符数量不超过 50。但是在一个问题的解中不允许出现空白字符和空行，相邻的两个问题的解之间用空行隔开。用大写字母 G 表示好人，大写字母 B 表示坏人。

解法：

思想：模拟实际过程，寻找前 n 个被“处死”的人的位置。

1. 普通解法——线性表“查找”法

1 用顺序存储结构实现

用数组记录当前所有未被处死的人原来的位置，初始值为 $1..2n$ 。可根据前一个被处死的人在数组中的位置（即下标）直接定位，找到下一个应该被处死的人在数组中的位置，然后删去，并将它后面的元素全部前移一次。

2 用链式存储结构实现

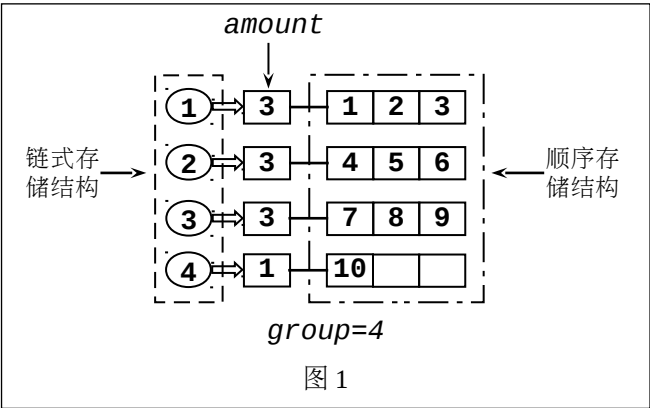
用链表记录当前所有未被处死的人原来的位置，初始值为 $1..2n$ 。每处死一个人后，只要将这个结点直接从链表中删去即可，然后指针后移 $(m-1)$ 次，找到下一个应该被处死的人。

2. 改进解法——“优化直接定位”法

总体思想就是在较好地实现“直接定位”的基础上，尽量避免大规模的元素移动。

设计出的数据结构如图 1 所示：其中 group 表示将原来的数据分为几段存储；每一段的开头记下的 amount 值表示此段中现有元素的个数。随程序的运行，amount 值是不断减小的。

这种结构可以看作是链式存储结构和顺序存储结构





的结合产物，兼具这两种存储结构的优点。运用了这种存储结构后，程序效率显著提高。

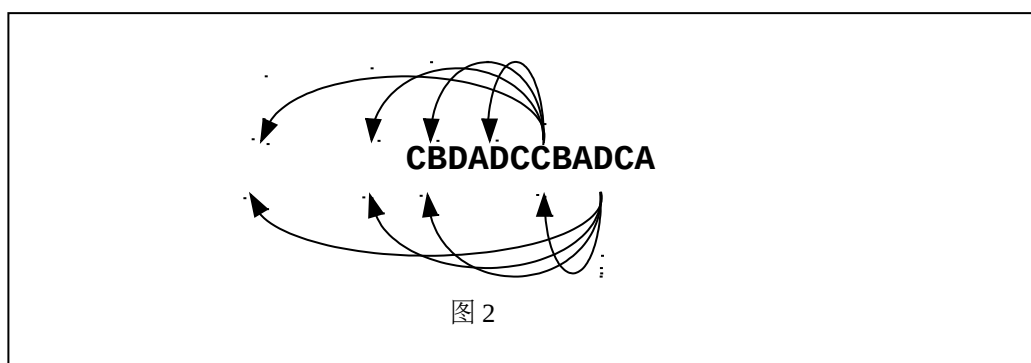
引申

➤ 横向延伸——其它约瑟夫环问题

如：《翻牌游戏》、《猴子选大王》

➤ 纵向延伸——数据结构的综合运用

在解决一些数据规模较大的题目时有很好的应用。如《隐藏的码字》(IOI'99)。在解决这道题目时，如果运用链式和顺序相结合的数据结构（如图 2 所示），程序效率就比较高。



链式和顺序相结合的数据结构实现简单，效果显著，应用比较广泛。当然还有其它的结合方式，比如二叉堆和顺序结构的一一映射（单射），在解决某些问题时有很好的效果。

小结

“网络式思维方式的核心是**联系**”。在做题目时，我们应深挖题目所给条件、各种数据结构以及算法之间的联系，这样才能更好地完成题目，并达到提高自己的目的。

这篇论文仅仅是从一类很常见的问题——约瑟夫环（也称 Josephus 排列）问题出发，并由此引申出数据结构的综合运用。对于形式多样的信息学问题来说，数据结构的综合运用只是解题策略中的一个小方面，但是如果我们对待每个问题、算法、数据结构等，都能深入发掘它与其它事物的联系，那么我们就可以自然而然地建立起知识网络，在必要的时候综合运用。而这对于我们的学习、研究将大有帮助。

特别需要强调的是：本文提到“数据结构的综合运用”，这里的综合并不单是指形式上的，更重要的是指思想（即内涵）的综合。只要在思想上体现出两种或多种数据结构的优点，在操作时发挥出它们的优点，就已经从根本上达到了综合的目的。