

Algorithm | BitMap & Sort |

比特图与排序

问题

- 算法教材与简要思路
- 200行代码
- 十几个函数
- 一周时间完成

- 为什么非要自己编写排序程序？
- 排序内容的内容，格式，总数？
- 为什么要在磁盘上排序？
- 其他特征？

- 最多一千万条记录
- 1MB左右可用内存
- 7位的正整数，且仅出现一次
- 无关联数据
- 输出升序排列的文件
- 理想输出时间为10秒左右

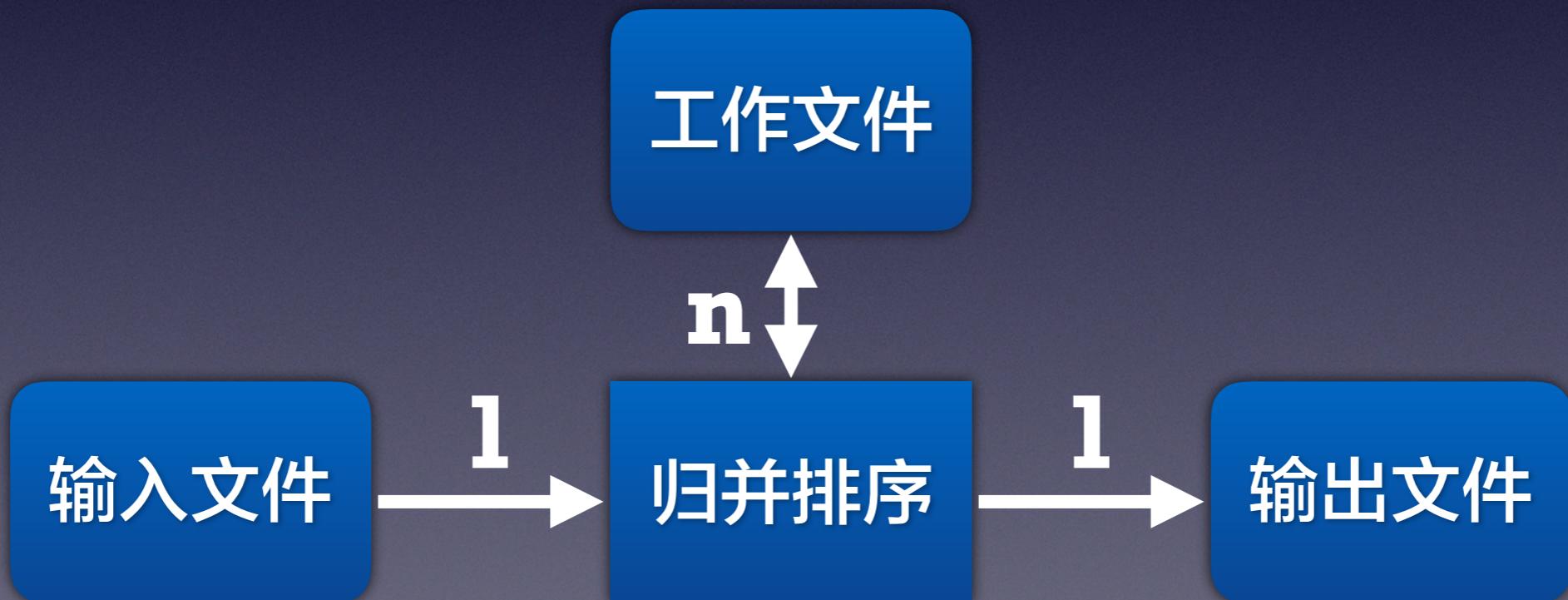
准确的问题描述

- 输入：一个最多包含 n 个正整数的文件，每个数都小于 n ， $n=10^7$ 。不重复，无关联
- 输出：按升序排列的整数列表
- 约束：1MB内存空间，磁盘空间不限。运行时间最多几分钟，10秒左右为佳

思考时间

归并排序

- 针对整数做调整，大约一百行代码。预计需要几天来完成



40趟算法

- 第一次读入0 - 249,999之间的整数，第二次读入250,000 - 499,999，以此类推
- 20行代码，不必考虑中间文件。



理想算法

?



**是否能够用大约800万
个可用位来表示最多
1,000万个互异的整数？**

1 MB \approx 1,000,000 Byte = 8,000,000 bit

实现概要

用20位的字符串来表示一个所有元素都小于20的简单正整数集合。

例: {1, 2, 3, 5, 8, 13}

0 1 1 1 0 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0

```
/* phase 1: initialize set to empty */
for i = [0, n]
    bit[i] = 0

/* phase 2: insert present elements into the set */
for each i in the input file
    bit[i] = 1

/* phase 3: write sorted output */
for i = [0, n]
    if bit[i] == 1
        write i on the output file
```

总结

- 正确的问题
- 位图数据结构
- 多趟算法
- 时间—空间折中与双赢
- 简单的设计

**完美的设计
= 不能再减少任何东西**

简单的程序 = 更可靠、更安全、更健壮、更高效，且易于实现与维护

