

动手解决生活中的实际问题

---

算法与生活

扑克牌整理

名字翻转

报名表排序

TOM的烦恼

### 成为最快的信息查询员

- ▶ 总共**10亿**条记录
- ▶ 没有数据库系统可用
- ▶ 希望能在最快的时间内查询出特定号码的信息

- ▶ `//依次读取每一条记录`
- ▶ `//判断号码是否相等`
- ▶ `//相等即返回信息并停止循环`

- ▶ 方法`get_time(message)`获得经过的时间
- ▶ 方法`write_to_temp_file(lines)`写入temp文件（8个）
- ▶ 方法`write_to_result_file()`读temp文件，排序，写入输出文件
- ▶ 方法`main`控制主流程



- ▶ `//定义每次获取文件的大小`
- ▶ `//定义一个循环，用来不间断的读取文件`
- ▶ `//调用写入temp文件的方法，将读取出来的内容传入`
- ▶ `//如果读取出的内容为空，就中断循环`

- ▶ `//获取经过了多少时间 (n = 当前时间 - 开始时间)`
- ▶ `//将输入的参数message添加到输出中 (例如: message + "in" + n + "seconds")`
- ▶ `//打印输出结果`

- ▶ **//定义号码的最大范围 (4375000000)**
- ▶ **//创建8个临时数组，并把传入的lines分别按照条件放入相应的数组**
- ▶ **//将数组存入临时文件**

- ▶ `//从小到大依次取出temp文件`
- ▶ `//对temp文件按照号码进行排序`
- ▶ `//写入result文件`

## DEBUG TIME

- ▶ 注意控制一次读取文件的量，别太大，也别太小
- ▶ 注意控制临时数组的取值范围，不要有遗漏也别有交集
- ▶ 思考一下如何取得32bit的最大值
- ▶ 小心考虑按号码排序的问题，思考按照字符串排序与按照整数类型排序的区别
- ▶ 良好的变量命名会为你节约很多的时间
- ▶ 注意\n陷阱。

- ▶ 方法`get_time(message)`获得经过的时间
- ▶ 方法`get_line(file)`获取特定的line
- ▶ 方法`final_search()`当范围极小时可以考虑采用顺序搜索（仅当你无法准确定位line时使用）
- ▶ 方法`search`控制主流程

## DEBUG TIME

- ▶ 如何定位磁盘文件上的一行?
- ▶ 当范围很小的时候如何控制文件读取的方向（向上读取或是向下读取?
- ▶ 当号码不存在时会发生什么?
- ▶ `string` 和 `int` 的比较如何进行? 如何“安全”的进行转换?
- ▶ 再一次，良好的变量命名会为你节约很多的时间

---

**PERFECTION IS ACHIEVED NOT WHEN  
THERE IS NOTHING MORE TO ADD, BUT  
WHEN THERE IS NOTHING LEFT TO TAKE  
AWAY**

**Antoine de Saint-Exupery**