

华东师范大学数据科学与工程学院上机实践报告

课程名称：算法设计与分析

指导教师：金澈清

上机实践名称：散列表

上机实践编号：No.7

年级：22级

姓名：唐健峰

学号10225501408

组号：408

上机实践成绩：

上机实践日期：

一、目的

- 1. 熟悉散列表的基本思想。
- 2. 掌握各种散列表的实现方法。

二、实验内容

- 1. 设计一个数据生成器，输入参数为 N；可以生成N个不重复的随机键或键值对。设计一个操作生成器，输入参数为N’，method；可以生成N’组操作method。操作包括插入和查询。
- 2. 基于开放寻址法实现哈希表及其插入和查询操作，选择合适的数据规模，计算在不同表的大小和不同已占用数量下的所需时间。
- 3. 以顺序访问的方式实现插入和查询。选择合适的数据规模，计算在不同表的大小和不同已占用数量下的所需的时间。
- 4. 对比散列表（哈希表）和顺序访问。
- 5. （思考题）探究不同散列函数对散列表性能的影响。

三、使用环境

推荐使用C/C++集成编译环境。

四. 实验过程

- 1. 写出数据生成器和两种算法的源代码。
- 2. 以合适的图表来表示你的实验数据。

```
#define RANGE 1000000 // 数字范围
#define HASH 100003 // 数组长度，即hash值
#define CIRCLE 5 // 循环取平均值
```

	10个数	100个数	1000个数	10000个数	100000个数
循序表	1.0815ms	15.3128ms	160.79ms	1825.25ms	32445.3ms
哈希表	0.0631ms	0.41887ms	3.74372ms	35.1656ms	354.859ms

```
#define RANGE 100000 // 数字范围
#define HASH 10003 // 数组长度，即hash值
#define CIRCLE 5~100 // 循环取平均值
```

令变量为产生的数量

	10个数	100个数	1000个数	10000个数
循序表	0.18669ms	2.13432ms	19.3807ms	455.535ms
哈希表	0.04596ms	0.28457ms	3.36533ms	34.5538ms

```
#define RANGE 10000 // 数字范围
#define HASH 1003 // 数组长度，即hash值
#define CIRCLE 5~1000 // 循环取平均值
```

	10个数	100个数	1000个数
循序表	0.0894ms	0.6322ms	5.605ms
哈希表	0.0764ms	0.4804ms	3.0994ms

五、总结

对上机实践结果进行分析，问题回答，上机的心得体会及改进意见。

在储存数字较小规模下，二者差异不是很明显，但随着数据储存的规模变大，哈希表明显比循序表好用，因为循序表会遍历整个数组直到找到元素，而哈希表可以提前算出数据大体范围，要改进的话我的方法是设计不同的哈希算法针对不同数据，采用更好的哈希算法之类的。