第八次实验报告

517021910499 陈鸣阳

说明：m=1 表示选择线性探测法 m=2 表示选择二次探测法
n=1 表示选择除留余数法
n=2 表示选择数字分析法
n=3 表示选择平方取中法
n=4 表示选择折叠法

Hash1.txt 的运行结果如下

```
the time of find:m=1n=1time:486.2ms
the time of insert:m=1n=1time:9960.6ms
the time of remove:m=1n=1time:244.8ms
the time of find:m=1n=2time:388.1ms
the time of insert:m=1n=2time:9889.2ms
the time of remove:m=1n=2time:193.8ms
the time of find:m=1n=3time:455.2ms
the time of insert:m=1n=3time:5453.2ms
the time of remove:m=1n=3time:223.8ms
the time of find:m=1n=4time:545.3ms
the time of insert:m=1n=4time:2217ms
the time of remove:m=1n=4time:282ms
the time of find:m=2n=1time:29.2ms
the time of insert:m=2n=1time:14848.8ms
the time of remove:m=2n=1time:14.2ms
the time of find:m=2n=2time:9.6ms
the time of insert:m=2n=2time:715.8ms
the time of remove:m=2n=2time:4.6ms
the time of find:m=2n=3time:49.6ms
the time of insert:m=2n=3time:4006.6ms
the time of remove:m=2n=3time:24.4ms
the time of find:m=2n=4time:11.7ms
the time of insert:m=2n=4time:3098.2ms
the time of remove:m=2n=4time:5.8ms
Process returned 0 (0x0)   execution time
Press any key to continue
```

查找：在查找操作上二次探测法有明显的优势，无论选取哪种散列函数，所需时间均比线性探测法快了一到两个数量级。其中二次探测法和数字分析法的组合用时最短。用时最长的是线性探测法和折叠法的组合。
插入：在插入操作上二次探测法的优势并不明显，因为二次探测法为了保证可靠性对数组长度有要求，所以插入过程中有个 doublespace 的操作导致耗时较长。其中用时最短的组合是二次探测法和数字分析法，用时最长的组合是二次探测法和除留余数法。
删除：对于删除操作二次探测法也具有明显的优势，因为删除操作的过程和查找其实是大致相同的，用时最短的组合是二次探测法和数字分析法。用时最长的组合是线性探测法和折叠法。

Hash2.txt 的运行结果如下：

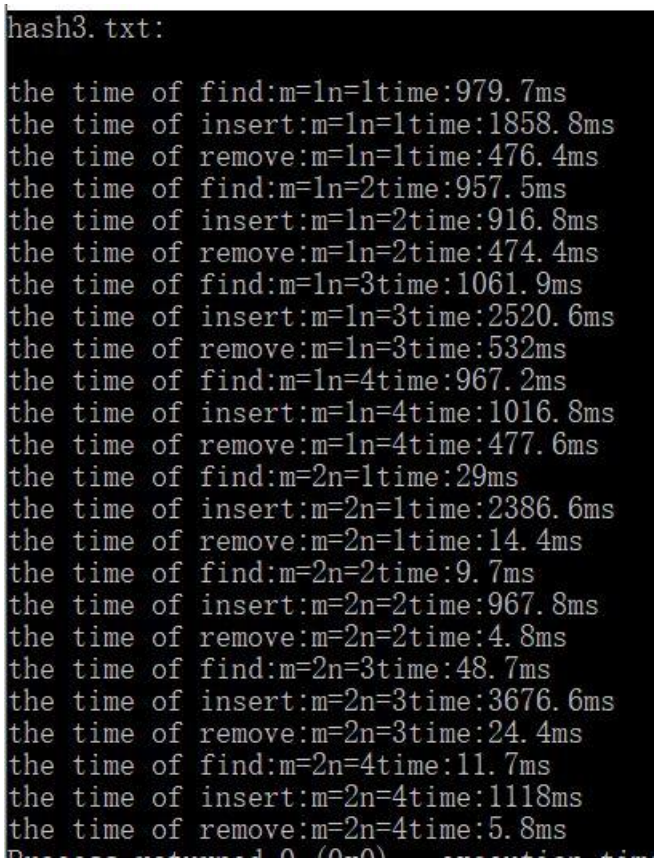
```
hash2.txt:

the time of find:m=1n=1time:235ms
the time of insert:m=1n=1time:10485.6ms
the time of remove:m=1n=1time:162.2ms
the time of find:m=1n=2time:393ms
the time of insert:m=1n=2time:880.6ms
the time of remove:m=1n=2time:212.8ms
the time of find:m=1n=3time:712.4ms
the time of insert:m=1n=3time:3537ms
the time of remove:m=1n=3time:331.8ms
the time of find:m=1n=4time:492.5ms
the time of insert:m=1n=4time:2323.4ms
the time of remove:m=1n=4time:176.4ms
the time of find:m=2n=1time:29.3ms
the time of insert:m=2n=1time:15608.6ms
the time of remove:m=2n=1time:14.2ms
the time of find:m=2n=2time:9.5ms
the time of insert:m=2n=2time:984.4ms
the time of remove:m=2n=2time:4.8ms
the time of find:m=2n=3time:48.4ms
the time of insert:m=2n=3time:4629.6ms
the time of remove:m=2n=3time:24.4ms
the time of find:m=2n=4time:11.6ms
the time of insert:m=2n=4time:3109.2ms
the time of remove:m=2n=4time:6ms
Process returned 0 (0x0)   execution tim
```

查找：对于不均匀的数据，在查找操作上二次探测法依旧有明显的优势。其中二次探测法和数字分析法的组合用时最短。用时最长的是线性探测法和平方取中法的组合。

插入：在插入操作上二次探测法的优势并不明显，因为二次探测法为了保证可靠性对数组长度有要求，所以插入过程中有个 doublespace 的操作导致耗时较长。其中用时最短的组合是线性探测法和数字分析法，用时最长的组合是二次探测法和除留余数法。

删除：对于删除操作二次探测法也具有明显的优势，因为删除操作的过程和查找其实是大致相同的，用时最短的组合是二次探测法和数字分析法。用时最长的组合是线性探测法和平凡取中法。

Hash3.txt 的运行结果如下：



查找：在查找操作上二次探测法依旧有明显的优势。其中二次探测法和数字分析法的组合用时最短。用时最长的是线性探测法和平方取中法的组合。

插入：在插入操作上二次探测法的优势并不明显，因为二次探测法为了保证可靠性对数组长度有要求，所以插入过程中有个 doublespace 的操作导致耗时较长。其中用时最短的组合是线性探测法和数字分析法，用时最长的组合是二次探测法和平方取中法。

删除：对于删除操作二次探测法也具有明显的优势，因为删除操作的过程和查找其实是大致相同的，用时最短的组合是二次探测法和数字分析法。用时最长的组合是线性探测法和平方取中法。

Hash4.txt 的运行结果如下：

```
hash4.txt:

the time of find:m=1n=1time:1318.8ms
the time of insert:m=1n=1time:1579.8ms
the time of remove:m=1n=1time:655ms
the time of find:m=1n=2time:1305.2ms
the time of insert:m=1n=2time:710.8ms
the time of remove:m=1n=2time:666.2ms
the time of find:m=1n=3time:857.6ms
the time of insert:m=1n=3time:2462.2ms
the time of remove:m=1n=3time:420.4ms
the time of find:m=1n=4time:1328.3ms
the time of insert:m=1n=4time:808.2ms
the time of remove:m=1n=4time:650.6ms
the time of find:m=2n=1time:28.9ms
the time of insert:m=2n=1time:2303ms
the time of remove:m=2n=1time:14.2ms
the time of find:m=2n=2time:9.7ms
the time of insert:m=2n=2time:961.4ms
the time of remove:m=2n=2time:5ms
the time of find:m=2n=3time:48.9ms
the time of insert:m=2n=3time:3611.8ms
the time of remove:m=2n=3time:24.6ms
the time of find:m=2n=4time:11.8ms
the time of insert:m=2n=4time:1114.4ms
the time of remove:m=2n=4time:6ms
```

查找：在查找操作上二次探测法依旧有明显的优势。其中二次探测法和数字分析法的组合用时最短。用时最长的是线性探测法和折叠法的组合。

插入：在插入操作上二次探测法的优势并不明显，因为二次探测法为了保证可靠性对数组长度有要求，所以插入过程中有个 doublespace 的操作导致耗时较长。其中用时最短的组合是线性探测法和数字分析法，用时最长的组合是二次探测法和平方取中法。

删除：对于删除操作二次探测法也具有明显的优势，因为删除操作的过程和查找其实是大致相同的，用时最短的组合是二次探测法和数字分析法。用时最长的组合是线性探测法和数字分析法。

从本次试验结果来看，数据均匀分布与不均匀分布并没有明显区别。

二次探测法除了在插入操作上表现略差于线性探测法，在查找和删除操作上均有明显优势，总体来看二次探测法的时间性能比线性探测法要好。

对于散列函数而言，平方取中法效率相对较低，数字分析法在多数情况下效率最高，除留余数法和折叠法表现一般。

综上，多数情况下，二次探测法和数字分析法的组合应该是一个较好的组合方式。