**稳定性测试**

随机生成十万个Key-Value对，先插入一半，再以随机顺序获取，查看Key对应的Value是否正确，然后删除30%，再随机获取，查看数据正确性。然后插入剩余的数据，再删除全部数据，每次都测试正确性。

**边界测试**

包括空表插入，全部删除，全部删除后插入，删除一半后插入。

**分支测试**

所有分支都走到。

**性能对比**

对比对象：红黑树（map），哈希表（unordered\_map）。

测试对象：基于内存的版本，基于磁盘日志的版本，基于共享内存的版本。

测试方法：1,000,000个KV对，进行插入，查找，删除十次，取平均时间，并统一量纲。对于共享内存，因涉及扩容，初始容量取1（最差），500,000（最好）。

测试结果：

|  |  |  |  |
| --- | --- | --- | --- |
| 数据结构名 | 平均插入时间 | 平均查找时间 | 平均删除时间 |
| 红黑树 | 10 | 4.5 | 6.8 |
| 哈希表 | 5 | 0.8 | 1.8 |
| 基于内存 | 6 | 3 | 4 |
| 基于磁盘日志 | 8 | 3 | 4 |
| 基于共享内存（最好） | 11 | 3.2 | 4.3 |
| 基于共享内存（最差） | 16 | 13 | 14 |

**内存占用**

经过测试，共享内存的内存占用约为其他结构的3倍（int型的Key和Value），Key和Value的数据结构越复杂（占用内存越多），共享内存占用空间较其他结构的比值越低，最好情况下，共享内存的占用等于其他数据结构（此时Key和Value的内存占用接近无限大）。