1. B树、存储模型
2. 字典树构造及其优化与应用
3. 持久化数据结构、序列化和反序列化的时机（java hibernate）
4. 大规模文本文件，全是单词，求前10词频的单词
5. 堆排序及其在求10词频问题中的应用
6. 字典树在统计词频上的的应用
7. 红黑树的特性及其在c++ stl中应用
8. 红黑树的调整
9. 贪心算法及其弊端
10. 能取得全局最优解的算法
11. 动态规划原理与本质
12. 01背包详细解释
13. 进程间通信
14. 数据库join类型与区别

inner join、left join、right join、cross join

1. 数据库ACID

原子性、一致性、隔离性、持久性

1. 实现bitmap数据结构，包括存储和插入方式
2. 实现unordered\_map，键为stirng，value不限
3. 实现unordered\_map过程中冲突的解决办法
4. 一串int型数组存放磁盘上的压缩存储方式，包括写入和读取及内存无法一次性读取时的解决办法
5. 对象复用
6. 零拷贝
7. 同步、异步、阻塞、非阻塞区别

同步异步表示消息的通知机制；

阻塞非阻塞表示等待消息通知时的状态；

消息通知的三种方式：状态、消息、回调

1. linux 的I/O模型
   1. 同步阻塞
   2. 同步非阻塞
   3. I/O复用
   4. 信号驱动
   5. 异步I/O
2. 异步I/O详细解释
3. strcut和class区别
4. stl库的介绍
5. vector使用的注意点及其原因
6. vector重新分配内存的大小与方式
7. hashmap的实现方式
8. map的实现方式
9. 红黑树的了解
10. c++虚函数的具体实现原理
11. 实现编译器处理虚函数表应该如何处理
12. 析构函数一般写成虚函数的原因
13. 解释哲学家进餐问题
14. 描述银行家算法
15. 实现一种算法解决哲学系进餐问题
16. 大数量整数去问题
17. 如果用bitmap解决大数量整数去重问题，计算当全为int型整数时需要消耗的内存
18. 环形公路加油站问题
19. 并查集
20. I/O复用技术，epoll优于select的原因
21. http中get和post的区别
22. 手写单例模式，线程安全

class A**{**

private**:**

static A**\*** a**;**

A**(**const A**&)=delete;**

A**&** **operator=(**const A**&)=delete;**

A**(){}**

public**:**

static A**\*** getSingle**(){**

**if(**a **==** **nullptr){**

lock**();**

**if(**a **==** **nullptr){**

a **=** **new** A**();**

**return** a**;**

**}**

unlock**();**

**}**

**return** a**;**

**}**

**}**

1. libevent的了解
2. 快排的稳定算法