

WXG二面

C++相关

- 重载以及重写（说反了）
 - 重载（Overload）是指不同的函数使用相同的函数名，但是函数的参数个数或类型不同。调用的时候根据函数的参数来区别不同的函数。
 - 重载是一个类中不同方法之间的一种关系
 - 覆盖/重写（Override）是指在派生类中重新对基类中的虚函数（注意是虚函数）重新实现。即函数名和参数都一样，只是函数的实现体不一样。
 - 父类方法，子类重写/覆盖父类方法
- i++/++i的效率（又反了）

```
class CInt
{
public:
    CInt& operator ++();
    CInt operator ++( int );
    CInt& operator +=( const CInt& that );
};

CInt operator +( const CInt& this, const CInt& that );
```

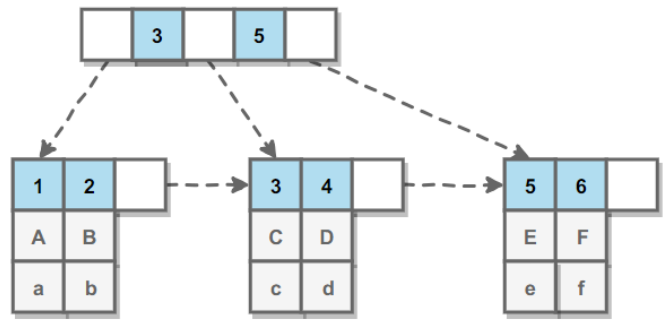
- `CInt& operator ++();` 对应 `++i`
 - 返回引用（返回++后的结果）的原因是因为在C++里，`++i`的结果应该是一个左值。对这个函数的调用，除了运算本身以外，并没有什么开销。（除了隐含的this以外，无传递参数，只有一个引用返回值，所以没有新的实例被创建）
- `CInt operator ++(int);` 对应 `i++`
 - 返回临时变量（++前的结果）
 - 这个函数中，会创建一个临时变量，并把它作为返回值拷贝给调用者，效率低

扩展：

- `CInt& operator +=(const CInt& that)`（对应于`i = i + j;`）传递一个参数，从理论上来说，它的开销和`++i`的开销是一样的，但是，如果你只是要对类的实例加一的话，应该用`++i`，因为那个函数可能为加一而特别优化过。
- `CInt operator +(const CInt& this, const CInt& that);` 对应于`k = i + j;`这个函数的开销与`i++`相同（需要创建一个临时变量），但是要注意的是，`i++`可能为加一而优化的。

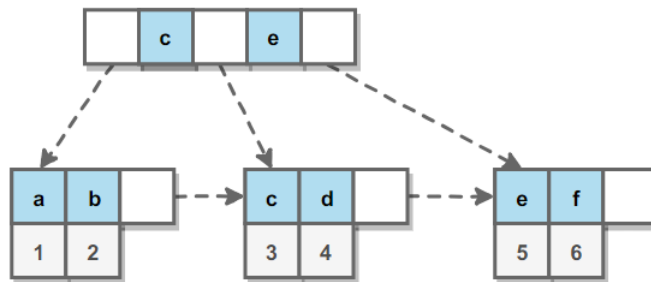
数据库

- 主索引和辅助索引的区别
 - 主索引项的排序方式和表中数据记录排序方式一致
 - 辅助索引中索引的逻辑顺序与磁盘上行的物理存储顺序不同
 - InnoDB 的 B+Tree 索引分为主索引和辅助索引。主索引的叶子节点 data 域记录着完整的数据记录，这种索引方式被称为聚簇索引。因为无法把数据行存放在两个不同的地方，所以一个表只能有一个聚簇索引。



CyC2018

辅助索引的叶子节点的 data 域记录着主键的值，因此在使用辅助索引进行查找时，需要先查找到主键值，然后再到主索引中进行查找。



CyC2018

算法

- 给n个数，求排序后的相邻数最大差值（只答对了一半）
 - 先找出最大值、最小值（确定桶范围）
 - 遍历数组置桶位true
- top k（最小k个数）
 - 维护一个大小为k的最大堆
 - 满堆时，检查堆顶，如果大于当前 num，堆顶换成 num，调用`sink`