C++研发高频笔/面试题目  
**笔试：**  
笔试这一块想不到特别好说的，很多时候只是公司筛选人的第一步，分数高低代表意义不大（尤其是大公司，现身说法，我的腾讯笔试真的是呵呵哒），当然笔试是一个门槛，不过后面很可能就不用谈了。我笔试经验也不足，感觉笔试考察最重点的就是算法的熟悉程度，代码能力，我笔试成绩通常不太高，原因是我编程题速度比较慢，往往都做不完。  
以C++开放举例，粗略说说笔试一般会考哪些内容：  
（1） 语言基础，C++中的定义，指针，模板，多态，重载等（尤其是多态，给定一个程序，基类子类有不同实现，考虑输出结果，这一类的题目面试题也极为高频）

**面试高频题：**  
校招过程中参考过牛客诸位大佬的面经，但是具体哪一块是参考谁的我也忘记了，如果您看到有些熟悉，那么可能就是您的，如果您觉得侵权，请告诉我，我撤回。  
1语言基础（C++）  
（1） 指针和引用的区别  
（2） 堆和栈的区别  
（3） new和delete是如何实现的，new 与 malloc的异同处  
（4） C和C++的区别  
（5） C++、Java的联系与区别，包括语言特性、垃圾回收、应用场景等（java的垃圾回收机制）  
（6） Struct和class的区别  
（7） define 和const的区别（编译阶段、安全性、内存占用等）  
（8） 在C++中const和static的用法（定义，用途）  
（9） const和static在类中使用的注意事项（定义、初始化和使用）  
（10） C++中的const类成员函数（用法和意义）  
（11） 计算下面几个类的大小：  
class A {};: sizeof(A) = 1;  
class A { virtual Fun(){} };: sizeof(A) = 4(32位机器)/8(64位机器);  
class A { static int a; };: sizeof(A) = 1;  
class A { int a; };: sizeof(A) = 4;  
class A { static int a; int b; };: sizeof(A) = 4;  
（12） 给一个代码，求输出结果  
class A  
{  
public:  
A(int x){}  
}  
问：A a = 1;是否正确, 如果正确, 那么它调用了哪些函数？  
这类题目更常见的是在基类和子类有不同实现方法。（虚函数相关，栗子很多，不多说了）  
（13） C++的STL介绍（这个系列也很重要，建议侯捷老师的这方面的书籍与视频），其中包括内存管理allocator，函数，实现机理，多线程实现等  
（14） STL源码中的hash表的实现  
（15） STL中unordered\_map和map的区别  
（16） STL中vector的实现  
（17） vector使用的注意点及其原因，频繁对vector调用push\_back()对性能的影响和原因。  
（18） C++中的重载和重写的区别：  
（19） C ++内存管理（热门问题）  
（20） 介绍面向对象的三大特性，并且举例说明每一个。  
（21） 多态的实现（和下个问题一起回答）  
（22） C++虚函数相关（虚函数表，虚函数指针），虚函数的实现原理（热门，重要）  
（23） 实现编译器处理虚函数表应该如何处理  
（24） 析构函数一般写成虚函数的原因  
（25） 构造函数为什么一般不定义为虚函数  
（26） 构造函数或者析构函数中调用虚函数会怎样  
（27） 纯虚函数  
（28） 静态绑定和动态绑定的介绍  
（29） 引用是否能实现动态绑定，为什么引用可以实现  
（30） 深拷贝和浅拷贝的区别（举例说明深拷贝的安全性）  
（31） 对象复用的了解，零拷贝的了解  
（32） 介绍C++所有的构造函数  
（33） 什么情况下会调用拷贝构造函数（三种情况）  
（34） 结构体内存对齐方式和为什么要进行内存对齐？  
（35） 内存泄露的定义，如何检测与避免？  
（36） 手写实现智能指针类（34-37我没遇见过）  
（37） 调试程序的方法  
（38） 遇到coredump要怎么调试  
（39） 内存检查工具的了解  
（40） 模板的用法与适用场景  
（41） 成员初始化列表的概念，为什么用成员初始化列表会快一些（性能优势）？  
（42） 用过C11吗，知道C11新特性吗？（有面试官建议熟悉C11）  
（43） C++的调用惯例（简单一点C++函数调用的压栈过程）  
（44） C++的四种强制转换

6设计模式和算法  
设计模式一般都不会考太多，除非你明确说自己懂。我基本上就不涉及到设计模式的东西，所以只是简要说说。当然，单例模式和简单工厂模式的概念和使用场景还是要知道的。  
（1） 介绍熟悉的设计模式（单例，简单工厂模式）  
（2） 写单例模式（饿汉模式和懒汉模式），线程安全版本  
（3） MVC设计模式  
算法这一块太过庞大，几乎都有可能，牛油们最好还是去刷剑指offer（level 1），leetcode（如果能够刷到最高难度，算法对你来说已经不是什么了，或者说面试对你来说简直就是吃饭喝水的难度），左神的书《程序源代码面试指南》（字符串，数组，dp，海量数据问题，搞定它们也就搞定面试的一半）。  
下面还是简单的列举一些吧（包括一些数据结构题目，只列举简单的，面试的算法一半不会太难，但是现在一般都是需要比较好的思维，或者曾经接触过这方面的题，建议就是多刷题，做题感觉是刷出来的）  
（1） 红黑树的了解（平衡树，二叉搜索树），使用场景  
（2） 红黑树在STL上的应用  
（3） 了解并查集吗？（低频）  
（4） 贪心算法和动态规划的区别  
（5） 判断一个链表是否有环，如何找到这个环的起点  
（6） 实现一个strcpy函数（或者memcpy），如果内存可能重叠呢  
（7） 实现一个循环队列  
（8） 排序算法（写快排，归并排序，堆排序），算法的时间复杂度，空间复杂度，是否稳定等  
（9） 快排存在的问题，如何优化  
（10） 反转一个链表  
（11） Top K问题（可以采取的方法有哪些，各自优点？）  
（12） Bitmap的使用，存储和插入方法  
（13） 二叉树的先序、中序、后序遍历（非递归实现）  
（14） 二叉树的公共祖先（简单地说，剑指offer上的题大都是高频题）  
（15） 1-n中有多少个1  
（16） 字典树的理解以及在统计上的应用  
（17） 数组的全排列  
（18） N个骰子出现和为m的概率  
（19） 海量数据问题（可参考左神的书）  
（20） 一致性哈希  
7智力题  
我特别把智力题单独拿出来讲，是因为这一块其实有些公司很看重（例如腾讯），但是我基本上也没刷过这类题目，接触不多，牛油们自行补充吧。  
（1） 100层楼，只有2个鸡蛋，想要判断出那一层刚好让鸡蛋碎掉，给出策略（滴滴笔试中两个铁球跟这个是一类题）  
（2） 毒药问题，n拼毒药，要快速找出哪一瓶有毒，需要几只小白鼠  
（3） 博弈论问题  
（4） 先手必胜策略问题：n本书，每次能够拿X-X本，怎么拿必胜  
（5） 放n只蚂蚁在一条树枝上，蚂蚁与蚂蚁之间碰到就各自往反方向走，问总距离或者时间。  
（6） 瓶子换饮料问题：多少个瓶子能够换1瓶饮料，问最多能喝几瓶