

## 字节跳动一面

1. 看你简历写得很简单。能简单的自我介绍一下你自己吗？

2. 对操作系统、数据库、网络技术有什么了解？

答：我们学校这些课程都是在大三下开设的，暂时对这方面的知识没有什么太大的了解。

3. 现在考察一下你的基本知识。你能说说C++中类和模板的联系与区别吗？

我心想：woc，这两个东西有联系吗？感觉不同的东西为啥要说区别呀？

答：类的话主要是对一些东西的抽象与封装，增强代码的复用性，同时隐藏一些实现细节。模板的话主要是对于常见的、公有的操作，但是进行抽象，这些操作往往和具体的类型是没有关系的，例如对于加法，都是差不多的，但是具体设计到，分为int的，float等等之类的。通过模板可以增强代码的复用性，避免重复编写相似代码，至于具体类型则可以通过模板实例化来确定。

感觉两者没有什么太多的联系和却别。可能是两者的交集，比如模板类，用于定义一些不依赖与具体类型的类。

4. 你能说一下C++的析构函数吗？析构函数是必须写的吗？

答：析构函数不是必须写。因为会有默认的析构函数。但是如果析构函数的成员有指针之类的，指针指向内存中某一个区域。则需要手动写析构函数，先将成员所指向的区域给析构掉。

另外，在这个时候面试官还提到了垃圾回收。让我感觉有点迷惑。因为我决定垃圾回收不是java之类的吗。让我摸不清面试官究竟是在问c++方面的还是java方面的。

5. 你能说一下你所了解到的一些设计模式吗？

我突然脑子瓦特，不记得有哪些模式，具体是咋样的。

答：比如说工厂模式，通过将类的构造器进行一定程度，例如包内隐藏，并创建相应的工厂类，然后对产生的类实例进行一些更好的限制或者提供更高的灵活性。另外，很多模式可以配合匿名类，使得代码更加灵活，更易于修改维护。

6. 你使用过c++中的vector之类的东西吗？

使用过。stl中的不少容器都使用过。

7. vector是线程安全的吗？

我迟疑了一下，因为我不知道面试官是在问c++中的vector还是java中的Vector.另外我的确没怎么主义多线程以及线程安全方面的问题。

答：vector是线程安全的。

事实是：c++中的vector不是线程安全的。Java中的Vector所有的公共方法单独使用是线程安全的，但是复合使用不是，需要手动加锁。

8. 你能说说红黑树、b树、b+树吗？

我内心方的一比。毕竟我真的不知道呀。

答: 这些都是平衡树。都是可以实现例如查找、插入、删除等操作的单次复杂度是 $O(\log n)$ 。其中b树、b+树是针对对一个节点的存取操作的代价非常昂贵的情况下使用的，诸如磁盘的IO就是比较昂贵。而红黑树一般是内存中，因为对一个节点的读取的速度更加快，那个时候，红黑树会优于b树、b+树。至于他们的具体操作，非常复杂，如果没有资料的话，我肯定是说不来也写不出来的。如果有资料的话倒是可以实现。（吹牛）

## 9. 那你能说一说它们的应用场景？说说他们的差异。

我现在回想起来他可能希望我回到map的实现是红黑树，然后b树、b+树在数据库中用到了。

答：因为它们都是平衡树。所以他们的应用场景是需要大量的插入、删除、更新、求前去后继之类的场景。差异的话，就是存储器是内存的时候红黑树更优。而b树和b+树我记得貌似是b+树相对于b树，兄弟之间多了边，更便于找一下个节点，更好遍历吧。

## 做题阶段

### 10. 基础知识我们理解的差不多了，那我们现在进入做题阶段吧。你能实现一个限流器吗？

我：（我一脸懵逼。没听清说的是啥）“xian什么？”

面：“限流器。就是xxxxxxx”（一句简单的解释）

我：（接着保持懵逼的表情）。

面：进一步的详细解释了一下限流器。“什么一个流量过来，多少ms，blablabla的……”大概两三分钟或者更久？没注意时间。

我：哦，我想一下。

面：嗯，你想一下！

我：（想了20s。心想，我还是不知道限流器是啥。）。额，就是xxxxxxx吗？

面：blablabla又说了一分钟的样子吧。

我：（想了30s的样子吧。）“额，我还是不是很理解我要写什么？”

面：额，可能现在这个东西对于大二的你还有点困难。你以前打比赛的时候有没有遇到这种有工程背景的题目，或者说纯粹是算法数据结构之类的题。

我：额，打比赛的时候没有碰到过有工程背景的题目。

### 11. 面：好，那这样我们就直接来看看算法方面的题目吧。

面试官开始口述题目，口述题目完毕之后，在侧边发出了题面的文字部分。

给定一个int数组A，数组中元素互不重复，给定一个数x，求所有求和能得到x的数字组合，组合中的元素来自A，

A = [10,1,2,7,6,5]

x = 8

结果：

[

[1, 7],

[1, 2, 5],

[2, 6],

[1, 1, 6]

[1,1,1,1,1,...]

]

我：额，首先比较简单的一种思路就是，对于第一个数，我们有两种选择。选择这个数，则变成了依旧是这个数组，但是x变成了x-A[0]的子问题；不选择这个数，则x不变，但是数组的元素去掉第一个数的子问题。但是这样可能复杂度会比较搞。我想一下能不能优化一下。

面：额，优化一下。

十几秒后

我：数组中所有数都是正数吗？

面：对，都是正数。

几十秒后

我：可以先对**A**排一下序，去重，然后枚举每一个数字使用了多少次。写成搜索就好了。

面: 那你写一下。

然后我开始写了。然而比较**sad**的是，我好久好久没写代码了。于是一个几十行程序我写了很久很久，最后给第二题只留下了**15**分钟的时间。以及我写出来的代码是不对的，有**bug**的。面试官可能看时间不怎么够了，所以让我对着已经写的代码解释。解释思路的时候我突然反应过来哪里出错了，然后就改了一下，跑出了可以看的结果。囧

```

#include <iostream>
#include <algorithm>
using namespace std;

void print(int a,int n) {
    for (int i = 0; i < n; ++i)
        cout<<a<<" ";
}

void print(int A[],int count[],int m) {
    cout<<"[";
    for (int i = 0; i < m; ++i)
        print(A[i],count[i]);
    cout<<"\n";
    /*if (i != m-1)
        cout<<res[i]<<" ";
    else
        cout<<res[i]<<"],\n";*/
}

bool work(int A[], int x,int n,int count[], int m) {
    if (x == 0) {
        print(A,count,m);
        return true;
    }
    if (x < 0 || n<=0) {
        return false;
    }
    for (int i = 0;i*A[m] <= x;++i) {
        count[m] = i;
        work(A,x-A[m]*i,n-1,count,m+1);
    }
    return false;
}

int A[]={10,1,2,7,6,5};
int n = 6;
int c[6];
int m;
int x = 8;

int main() {

    sort(A,A+n);
    int cnt =unique(A,A+n)-A;
    work(A,x,cnt,c,0);
    return 0;
}

```

12. 面：下面我们看一下第二题。blabla一通。然后给出文字部分的题目如下。

给定1个二维字符数组m和单词1个w，搜索w是否在m中。搜索的定义是从m的任意位置开始，可以上下左右移动，依次

```
a  c  d  z
x  t  r  o
f   i  o  o
```

我： 额，意思是这个单词可以在这个二维字符数组中转弯？

面： 对，可以转弯。现在时间也只有十五分钟了，写肯定是写不完了。要不你就说说思路吧。

我： 一种比较简单的思路是搜索。通过首先在字符数组中找到这个单词第一个字符在字符数组中的位置，然后往四个方向拓展，进行广搜。不过广搜的话可能占用的空间会比较大，或许深搜dfs会好一点。毕竟只是判断存不存在。不过我感觉复杂度会比较高。（我思考了30s，发现我并没有更好的方案了，所以不在说话。）

面： 嗯，Dxxx(没听清是啥)肯定是可以的。你都想到这个了，写肯定是能写出来。现在时间不多了，就不写了。

13. 面: 你算法方面有了解过**机器学习方面的算法**吗？

答: 没有。

14. 你还有什么问题需要问我的吗？