**C/C++/STL/Python(熟练)**

1. 静态成员

解析：类的静态成员属于整个类 而不是某个对象，可以被类的所有方法访问，子类当然可以父类静态成员；静态成员无多态性！静态方法属于整个类，在对象创建之前就已经分配空间，类的非静态成员要在对象创建后才有内存，所有静态方法只能访问静态成员，不能访问非静态成员；静态成员可以被任一对象修改，修改后的值可以被所有对象共享。

静态成员可以实现多个对象之间的数据共享。

静态成员函数和静态数据成员一样，它们都属于类的静态成员，它们都不是对象成员。因此，对静态成员的引用不需要用对象名。

在静态成员函数的实现中不能直接引用类中说明的非静态成员，可以引用类中说明的静态成员（这点非常重要）。如果静态成员函数中要引用非静态成员时，可通过对象来引用。从中可看出，调用静态成员函数使用如下格式：<类名>::<静态成员函数名>(<参数表>);

1. 多态，虚函数理解（虚表指针）
2. STL迭代器失效
3. 字节对齐

解析：对于**单层结构体**要遵循下面2个原则:

（1）结构体变量中每个成员的偏移量都要是其自身大小的整数倍（第一个成员直接不用看可以,因为其偏移量为0,而任何数的0倍都是0）,不是整数倍要在上一个成员后面补字节.

（2）结构体大小(包含补上的字节大小)必须是所有成员大小(成员自身大小,不包含在其后面补上的字节)的整数倍，也即所有成员大小的公倍数。不是整数倍要在最后一个成员后面补字节

对于**嵌套的结构体**，**需要将其展开**。对结构体求sizeof时，上述两种原则变为：

（1）展开的子结构体中的第一个成员的偏移量应当是父结构体中除子结构体外最大成员大小的整数倍。

（2）结构体大小必须是所有成员大小的整数倍，这里所有成员计算的是展开后的成员，而不是将子结构体当做一个整体。

Sizeof()类，虚函数？？

1. 指针++：

main()

{

    char\*a[]={"work","at","alibaba"};

    char\*\*pa=a;

    pa++;

    printf("%s",\*pa); //at

}

1. C/C++四种类型转换

**数据结构+图像处理+机器学习和深度学习在计算机视觉中的应用（熟练）**

数据结构：

1. 链表：（链表反转，插入排序，归并排序，）
2. 栈：（辅助实现DFS，Get Max Stack, 两个栈实现队列，Sort Stack, Validate parenthesis,二叉树中序遍历，Evaluate Expression,环判断）：
3. 队列：（生产者-消费者,Queue with Max）
4. 树的先，中，后非递归遍历
5. 二叉排序树的查找，插入，删除操作
6. 二叉排序树节点的下一个节点
7. 平衡二叉树，查找，插入，删除操作
8. B-/+树基本概念，查找，插入，删除操作

（B树：二叉树，每个结点只存储一个关键字，等于则命中，小于走左结点，大于走右结点；

B-树：多路搜索树，每个结点存储M/2到M个关键字，非叶子结点存储指向关键字范围的子结点；所有关键字在整颗树中出现，且只出现一次，非叶子结点可以命中；

B+树：在B-树基础上，为叶子结点增加链表指针，所有关键字都在叶子结点中出现，非叶子结点作为叶子结点的索引；B+树总是到叶子结点才命中；

B\*树：在B+树基础上，为非叶子结点也增加链表指针，将结点的最低利用率从1/2提高到2/3；1/3的数据到新结点，最后在父结点增加新结点的指针；所以，B\*树分配新结点的概率比B+树要低，空间使用率更高。）

1. Trie树（字典树）基本概念
2. Hash表，哈希函数和冲突处理方法
3. Dijkstra算法

图像处理：

1. 对比度拉伸，直方图均衡，直方图匹配

2. 仿射变换，透视变换

3. 连通域求解

机器学习： https://www.zhihu.com/question/59683332/answer/281642849

1.SVM理解

2.LR推导

3.最小二乘法

深度学习：

1. BP算法
2. 各类优化算法

**操作系统+数据库+网络+linux（知道重要的）**

1. 页面置换算法
2. TCP/IP协议

解析：TCP建立连接时，首先客户端和服务器处于close状态。然后客户端发送SYN同步位，此时客户端处于SYN-SEND状态，服务器处于lISTEN状态，当服务器收到SYN以后，向客户端发送同步位SYN和确认码ACK，然后服务器变为SYN-RCVD，客户端收到服务器发来的SYN和ACK后，客户端的状态变成ESTABLISHED(已建立连接)，客户端再向服务器发送ACK确认码，服务器接收到以后也变成ESTABLISHED。然后服务器客户端开始数据传输

1. linux的I/O复用接口select和epoll: <http://blog.csdn.net/tennysonsky/article/details/45745887>

（1）代码题（leetcode类型），主要考察数据结构和基础算法，以及代码基本功

虽然这部分跟机器学习，深度学习关系不大，但也是面试的重中之重。基本每家公司的面试都问了大量的算法题和代码题，即使是商汤、face++这样的深度学习公司，考察这部分的时间也占到了我很多轮面试的60%甚至70%以上。我去face++面试的时候，面试官是residual net，shuffle net的作者；但他们的面试中，写代码题依旧是主要的部分。

大部分题目都不难，基本是leetcode medium的难度。但是要求在现场白板编程，思路要流畅，能做到一次性Bug-free. 并且，一般都是要给出时间复杂度和空间复杂度最优的做法。对于少数难度很大的题，也不要慌张。一般也不会一点思路也没有，尽力给面试官展现自己的思考过程。面试官也会引导你，给一点小提示，沿着提示把题目慢慢做出来也是可以通过面试的。

以下是我所遇到的一些需要当场写出完整代码的题目：

<1> 二分查找。分别实现C++中的lower\_bound和upper\_bound.

<2> 排序。 手写快速排序，归并排序，堆排序都被问到过。

<3> 给你一个数组，求这个数组的最大子段积

时间复杂度可以到O(n)

<4> 给你一个数组，在这个数组中找出不重合的两段，让这两段的字段和的差的绝对值最大。

时间复杂度可以到O(n)

<5> 给你一个数组，求一个k值，使得前k个数的方差 + 后面n-k个数的方差最小

时间复杂度可以到O(n)

<6> 给你一个只由0和1组成的字符串，找一个最长的子串，要求这个子串里面0和1的数目相等。

时间复杂度可以到O(n)

<7> 给你一个数组以及一个数K， 从这个数组里面选择三个数，使得三个数的和小于等于K， 问有多少种选择的方法？

时间复杂度可以到O(n^2)

<8> 给你一个只由0和1组成的矩阵，找出一个最大的子矩阵，要求这个子矩阵是方阵，并且这个子矩阵的所有元素为1

时间复杂度可以到O(n^2)

<9> 求一个字符串的最长回文子串

时间复杂度可以到O(n) (Manacher算法)

<10> 在一个数轴上移动，初始在0点，现在要到给定的某一个x点， 每一步有三种选择，坐标加1，坐标减1，坐标乘以2，请问最少需要多少步从0点到x点。

<11> 给你一个集合，输出这个集合的所有子集。

<12> 给你一个长度为n的数组，以及一个k值（k < n) 求出这个数组中每k个相邻元素里面的最大值。其实也就是一个一维的max pooling

时间复杂度可以到O(n)

<13> 写一个程序，在单位球面上随机取点，也就是说保证随机取到的点是均匀的。

<14> 给你一个长度为n的字符串s，以及m个短串（每个短串的长度小于10）， 每个字符串都是基因序列，也就是说只含有A,T,C,G这四个字母。在字符串中找出所有可以和任何一个短串模糊匹配的子串。模糊匹配的定义，两个字符串长度相等，并且至多有两个字符不一样，那么我们就可以说这两个字符串是模糊匹配的。

<15> 其它一些描述很复杂的题这里就不列了。

（2）数学题或者"智力"题。

不会涉及特别高深的数学知识，一般就是工科数学（微积分，概率论，线性代数）和一些组合数学的问题。

下面是我在面试中被问到过的问题：

<1> 如果一个女生说她集齐了十二个星座的前男友，她前男友数量的期望是多少？

ps：这道题在知乎上有广泛的讨论，作为知乎重度用户我也看到过。[如果一个女生说，她集齐了十二个星座的前男友，我们应该如何估计她前男友的数量？](https://www.zhihu.com/question/38331955" \t "_blank)

<2> 两个人玩游戏。有n堆石头，每堆分别有a1, a2, a3.... an个石头，每次一个游戏者可以从任意一堆石头里拿走至少一个石头，也可以整堆拿走，但不能从多堆石头里面拿。无法拿石头的游戏者输，请问这个游戏是否有先手必胜或者后手必胜的策略？ 如果有，请说出这个策略，并证明这个策略能保证必胜。

<3> 一个一维数轴，起始点在原点。每次向左或者向右走一步，概率都是0.5. 请问回到原点的步数期望是多少？

<4> 一条长度为1的线段，随机剪两刀，求有一根大于0.5的概率。

<5> 讲一下你理解的矩阵的秩。低秩矩阵有什么特点？ 在图像处理领域，这些特点有什么应用？

<6> 讲一下你理解的特征值和特征向量。

<7> 为什么负梯度方向是使函数值下降最快的方向？简单数学推导一下

（3）机器学习基础

这部分建议参考周志华老师的《机器学习》。

下面是我在面试中被问到过的问题：

<1> 逻辑回归和线性回归对比有什么优点？

<2> 逻辑回归可以处理非线性问题吗？

<3> 分类问题有哪些评价指标？每种的适用场景。

<4> 讲一下正则化，L1和L2正则化各自的特点和适用场景。

<5> 讲一下常用的损失函数以及各自的适用场景。

<6> 讲一下决策树和随机森林

<7> 讲一下GBDT的细节，写出GBDT的目标函数。 GBDT和Adaboost的区别与联系

<8> 手推softmax loss公式

<9> 讲一下SVM, SVM与LR有什么联系。

<10>讲一下PCA的步骤。PCA和SVD的区别和联系

<11> 讲一下ensemble

<12> 偏差和方差的区别。ensemble的方法中哪些是降低偏差，哪些是降低方差？

...... 这部分问得太琐碎了，我能记起来的问题就这么多了。我的感觉，这部分问题大多数不是问得很深，所以不至于被问得哑口无言，总有得扯；但是要想给出一个特别深刻的回答，还是需要对机器学习的基础算法了解比较透彻。

（4）深度学习基础

这部分的准备，我推荐花书（Bengio的Deep learning）和 [@魏秀参](https://www.zhihu.com/people/b716bc76c2990cd06dae2f9c1f984e6d) 学长的[《解析卷积神经网络-深度学习实践手册》](http://210.28.132.67/weixs/book/CNN_book.html)

下面是我在面试中被问到过的问题：

<1> 手推BP

<2> 手推RNN和LSTM结构

<3> LSTM中每个gate的作用是什么，为什么跟RNN比起来，LSTM可以防止梯度消失

<4> 讲一下pooling的作用， 为什么max pooling要更常用？哪些情况下，average pooling比max pooling更合适？

<5> 梯度消失和梯度爆炸的原因是什么？ 有哪些解决方法？

<6> CNN和RNN的梯度消失是一样的吗？

<6> 有哪些防止过拟合的方法？

<7> 讲一下激活函数sigmoid，tanh，relu. 各自的优点和适用场景？

<8> relu的负半轴导数都是0，这部分产生的梯度消失怎么办？

<9> batch size对收敛速度的影响。

<10> 讲一下batch normalization

<11> CNN做卷积运算的复杂度。如果一个CNN网络的输入channel数目和卷积核数目都减半，总的计算量变为原来的多少？

<12> 讲一下AlexNet的具体结构，每层的作用

<13> 讲一下你怎么理解dropout，分别从bagging和正则化的角度

<14> data augmentation有哪些技巧？

<15> 讲一下你了解的优化方法，sgd, momentum, rmsprop, adam的区别和联系

<16> 如果训练的神经网络不收敛，可能有哪些原因？

<17> 说一下你理解的卷积核， 1x1的卷积核有什么作用？

........

同上，这部分的很多问题也是每个人都或多或少能回答一点，但要答得很好还是需要功底的。

（5）科研上的开放性问题

这部分的问题没有固定答案，也没法很好地针对性准备。功在平时，多读paper多思考，注意培养自己的insight和intuition

下面是我在面试中被问到过的问题：

<1> 选一个计算机视觉、深度学习、机器学习的子领域，讲一下这个领域的发展脉络，重点讲出各种新方法提出时的motivation，以及谈谈这个领域以后会怎么发展。

<2> 讲一下你最近看的印象比较深的paper

<3> 讲一下经典的几种网络结构， AlexNet， VGG，GoogleNet， Residual Net等等，它们各自最重要的contribution

<4> 你看过最近很火的XXX paper吗? 你对这个有什么看法？

......

（6） 编程语言、操作系统等方面的一些问题。

C++， Python， 操作系统，Linux命令等等。这部分问得比较少，但还是有的，不具体列了

（7）针对简历里项目/论文 / 实习的一些问题。

这部分因人而异，我个人的对大家也没参考价值，也不列了。