

[返回目录:全网各大厂 iOS 面试题-题集大全](#)

## iOS 面试-最新快手 iOS 面试题

### 背景

面的是快手 X3 岗位，视频面试，不支持周末，但是可以选择晚上时间。视频面试是通过牛客网进行的，以下是记下来的各轮面试题，对于一些 iOS 基础知识就不做解答了。

### 一面

1、用递归写一个算法，计算从 1 到 100 的和。

```
func sum(value: Int) -> Int {
    if value <= 0 {
        return 0
    }
    var number = value
    return value + sum(value: number - 1)
}
// 计算过程
let result = sum(value: 100)
print(result)
复制代码
```

写完算法之后又围绕着问了几个问题，都是算法基础：

- 算法的时间复杂度是多少
- 递归会有什么缺点
- 不用递归能否实现，复杂度能否降到  $O(1)$

2、property 的作用是什么，有哪些关键词，分别是什么含义？

3、父类的 property 是如何查找的？

4、NSArray、NSDictionary 应该如何选关键词？

5、copy 和 mutableCopy 有什么区别，深复制和浅复制是什么意思，如何实现深复制？

6、用 runtime 做过什么事情？runtime 中的方法交换是如何实现的？

7、讲一下对 KVC 和 KVO 的了解，KVC 是否会调用 setter 方法？

8、\_\_block 有什么作用

9、说一下对 GCD 的了解，它有那些方法，分别是做什么用的？

10、对二叉树是否了解？

面试官是想接着问这方面的问题的。我当时说了不了解，然后就没有后续了。

## 二面

1、ARC 和 MRC 的区别，iOS 是如何管理引用计数的，什么情况下引用计数加 1 什么情况引用计数减一？

2、在 MRC 下执行[object autorelease]会发生什么，autorelease 是如何实现的？

3、OC 如何实现多继承？

这个当时没有答好。其实借助于消息转发，protocol 和类别都可以间接实现多继承。

4、对设计模式有什么了解，讲一下其中一种是如何使用的。

5、有没有哪个开源库让你用的很舒服，讲一下让你舒服的地方。

6、一张 100\*100，RGBA 的 png 图像解压之后占多大内存空间。

5、算法题

题目：给定一个个数字 arr，判断数组 arr 中是否所有的数字都只出现过一次。

这个并没有要求写出来，说是提供思路就行了。我当时给的方案是在便利数组的时候，用一个字典把便利的元素存起来，如果在后面的便利过程中新元素在字典中存在过就说明，有重复数字出现。时间复杂度是  $O(n)$ 。

当时也问了有没有办法进行优化，我当时想到了将数组转成 Set，然后和原数组比较，两个集合的数量是否变化。

7、因为我跟他介绍自己 Swift 用的多一些，然后问了些 Swift 跟 OC 的区别，各自的优缺点。

8、为什么离职，有什么职业规划。

## 三面

1、给定一个 Int 型数组，用里面的元素组成一个最大数，因为数字可能非常大，用字符串输出。

输入：[3,30,34,5,9]

输出：9534330

复制代码

这个是 leetcode 的 179 题，难度中等。面试官让先说思路，再去做题。事先说一下这个题我没有做过。当时的思路是用冒泡法进行排序，排序的前提是将较少位数的数字进行循环补齐，例如 3 和 30 的比较，变成 33 和 30 的比较，34 和 4 的比较变成 34 和 44 的比较，然后将结果从大到小整合成字符串输出。

但是做题是却发现没那么简单，位数的补齐对于 2 位和 3 位数的比较还要求位数的最小公倍数，将他们都转成 6 位数才能比较。在挣扎了 5 分钟做了就做罢了。

后来再去做这道题，其实这就是一个排序而已，只不过他的规则是按高位优先级更高的原则，而这一点跟字符串的比较保持一致，如果再加一些 Swift 的高阶函数，就可以写成：

```
func largestNumber(_ nums: [Int]) -> String {
    let sort = nums.map {"\($0)"}.sorted { (lStr, rStr) -> Bool in
        return lStr + rStr > rStr + lStr
    }
    let result = sort.joined()
    if result.prefix(1) == "0" {
        return "0"
    } else {
        return result
    }
}
```

复制代码

执行结果： **通过** [显示详情 >](#)

执行用时： **16 ms**，在所有 Swift 提交中击败了 **100.00%** 的用户

内存消耗： **20.9 MB**，在所有 Swift 提交中击败了 **22.22%** 的用户

2、项目中有这么一个方法 `func findfile(dir: String suffix: String) -> [String]`，可以通过输入文件夹目录，和后缀检索出所需的文件。

例如需要在某个文件中检索 txt 文件或者 mp4 文件，那就传入 dir 和 suffix 就行了。现在又有一些需求，例如需要检索 utf8 格式的 txt 或者 h264 编码的 mp4，也会有一些例如查找最近一周更新过的文件这样的需求，你如何优化这个类，让它满足这些情况？

我首先想到的是这么多需求不可能一个方法就完成，需要根据不同场景拆出不同的方法，但是这些同属于文件操作，会有一个共同使用的方法就是检索文件。这个方法需要传入文件目录，然后递归的返回当前目录所有文件路径。外部不同场景的调用逻辑就用一个 enum 完成，不同值对应相同范围的不同种类。

面试官比较关注内部共用的文件检索怎么写，他说子文件如果过多怎么办，如何优化。我有点懵，查找文件至少是要遍历一遍的，子文件过多，这个应该是没法优化的啊。中间卡了一段时间，后来他给了提示说是不是可以用 **block** 实现，将文件路径返回出去，由外部决定当前文件是否可用，最终外部的调用类是这个样子。

```
//我的方案
//func findDir(_ dir: String) -> [String]
//block 方案
func findDir(_ dir: String, block: ((String) -> Bool))
复制代码
```

我想来确实没毛病，用 **block** 返回内容至少不会将该目录的所有文件都由一个对象持有，而前面一堆的铺垫其实也都是为验证 **block** 方案的好处。

其实事后想下这个问题没啥难的，这种写法自己也有写过，但当时就是没想起来，可能前面一圈的铺垫给我带偏了吧，说亏也不亏，以后多多努力吧。

## 总结

整体来看，快手的面试题跟我在别处看到的 iOS 面试题对比要简单些。一面主要是基础知识，二面考察更全面一些，更多让自己谈一些对技术的理解，三面则是更偏实践一些。

算法虽然三轮都有，但相对比较简单，即使写不出来，有思路也是可以的。当然写出来肯定是加分项，所以大家准备面试时，应该都看一下。算法相关的，排序，数组，二叉树，这几类是重点。

## 返回目录:全网各大厂 iOS 面试题-题集大全

---

### 更多精选大厂 · iOS 面试题答案 PDF 文集



获取加小编的 iOS 技术交流圈: [937 194 184](#), 直接获取