

GeekBand 极客班

互联网人才加油站!

# 算法入门和编程风格

[www.geekband.com](http://www.geekband.com)

## GeekBand 极客班 互联网人才+加油站！

极客班携手网易云课堂，针对热门IT互联网岗位，联合业内专家大牛，紧贴企业实际需求，量身打造精品实战课程。

### 专业课程 + 项目碾压

- 顶尖专家技能私授
- 贴合企业实际需求
- 互动交流直播答疑
- 学员混搭线上组队
- 一线项目实战操练
- 业内大牛辅导点评



[www.geekband.com](http://www.geekband.com)

# 算法入门和编程风格

GeekBand 极客班

## 讲师简介

主讲 董飞:

LinkedIn面试官，三年面试经验，面试人数150+，国内外 Offer 10+，丰富的国内+海外工作经验。

# 大纲

从一道入门题说起

面试中常见的误区

如何准备面试算法

了解算法面试中的模板

面试常见问题答疑

GeekBand

极客班

# 从一道入门题说起

GeekBand

极客班

# StrStr

**`/* Returns the position of the first occurrence of string target in string source, or -1 if target is not part of source.*/`**

GeekBand

```
1 class Solution {
2     /**
3      * Returns a index to the first occurrence of target in source, or -1 if
4      * @param source string to be scanned.
5      * @param target string containing the sequence of characters to match.
6      */
7     public int strStr(String source, String target) {
8         if (source == null || target == null) {
9             return -1;
10        }
11
12        int i, j;
13        for (i = 0; i < source.length() - target.length() + 1; i++) {
14            for (j = 0; j < target.length(); j++) {
15                if (source.charAt(i + j) != target.charAt(j)) {
16                    break;
17                } // if
18            } // for j
19            if (j == target.length()) {
20                return i;
21            }
22        } // for i
23
24        return -1;
25    }
26 }
```



# StrStr常见问题1

我知道一个算法叫KMP

GeekBand

极客班

## StrStr常见问题2

```
public String strStr(String s1, String s2) {  
    if (s1==null || s2==null)  
        return -1;  
    int j;  
    for (int i=0;i<s1.length()-s2.length()+1;i++) {  
        for (j=0;j<s2.length();j++)  
            if (s1.charAt(i+j)!=s2.charAt(j))  
                break;  
        if (j==s2.length())  
            return i;  
    }  
    return -1;  
}
```

## strStr常见问题3

```
public String strStr(String src, String dest) {  
    int i, j;  
    for (i = 0; i < src.length(); i++) {  
        for (j = 0; j < dest.length(); j++) {  
            if (src.charAt(i + j) != dest.charAt(j)) {  
                break;  
            }  
        }  
        if (j == dest.length()) {  
            return src.substring(i);  
        }  
    }  
    return null;  
}
```

## StrStr常见问题4

Java:

```
for (i = 0; i < source.length() - target.length(); i++)
```

C++:

```
for (i = 0; i < strlen(source) - strlen(target); i++)
```

# 面试中的常见误区

GeekBand 极客班

## 面试中的常见误区

做过的题(或者简单的题)肯定能过

算法想出来了就能过

代码写出来了就能过

GeekBand

极客班

# 面试官眼中的求职者

你可能是他未来的同事

你的代码看起来舒服么？

TA需要多少时间Review你的代码

你的Coding习惯好么？

TA不会在未来疲于帮你DEBUG，你不会经常就搞出事故

你的沟通能力好么？

TA和你交流费劲么

# 面试考察的编程基本功

程序风格（缩进，括号，变量名）

Coding习惯（异常检查，边界处理）

沟通（让面试官时刻明白你的意图）

测试（主动写出合理的Testcase）



# 你真的会面试么？

你做题之前，先在白纸上写一遍么？

写了200多题？你吃透了几题？

题目不会直接说不会么？

是不是觉得面试官在为难你？

GeekBand

极客班

# 如何练习算法

GeekBand 极客班

# 算法，永远的痛

题做了很多，但就是记不住解法

从来就没有弄明白过动态规划是怎么回事

这题好像见过，不过还是不知道怎么做

leetcode, cc150都刷了，新题还是跪

Geekband

极客班

# 代码风格

代码块可为三大块：异常处理（空串和边界处理），主体，返回

代码风格(可参考Google的编程语言规范)

- 变量名的命名(有意义的变量名)
- 缩进(语句块)
- 空格(运算符两边)
- 代码可读性(即使if语句只有一句也要加花括号)

《代码大全》中给出的参考

# 基本代码素养

关于空格

用空行把大块代码分成逻辑上的“段落”

关于括号

C 指针中的指针符靠近类型名，如写成 `int* p`，而不写成 `int *p`

一个函数只专注做一件事

关于命名

# 实战算法策略

总结归类相似题目

找出适合同一类题目的模板程序

对基础题熟练掌握

GeekBand

极客班

再看一道简单题

# Memmove

```
void *memmove(void *dest, const void *src, size_t n)
{
    // implementation here
}
```



# 陷阱

- 内存重叠的处理
- 临时变量太多或者没安全释放
- 没有测试内存越界
- 指针操作不熟悉

```
void *memmove(void *dest, const void *src,
size_t n)
{
    char *p1 = dest;
    char *p2 = src;

    while (*p2 != \0)
        *p1++ = *p2++;

    return p1;
}
```



## 正确写法

```
void *memmove(void *dest, const void *src, size_t n)
{
    char *p1 = dest;
    const char *p2 = src;

    if (p2 < p1) {
        p2 += n;
        p1 += n;
        while (n-- != 0)
            *--p1 = *--p2;
    } else {
        while (n-- != 0)
            *p1++ = *p2++;
    }
    return p1;
}
```

# 排列组合模板

GeekBand 极客班

# Subsets

$\{1,2,3\}$

$\{\}, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}$

GeekBand

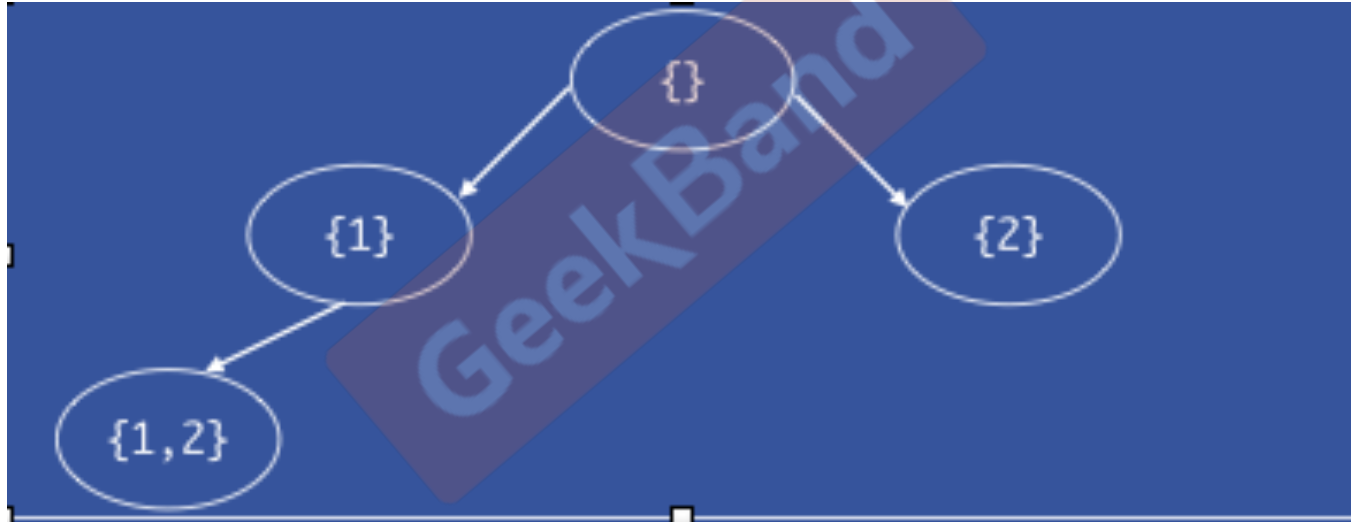
极客班

# Subsets

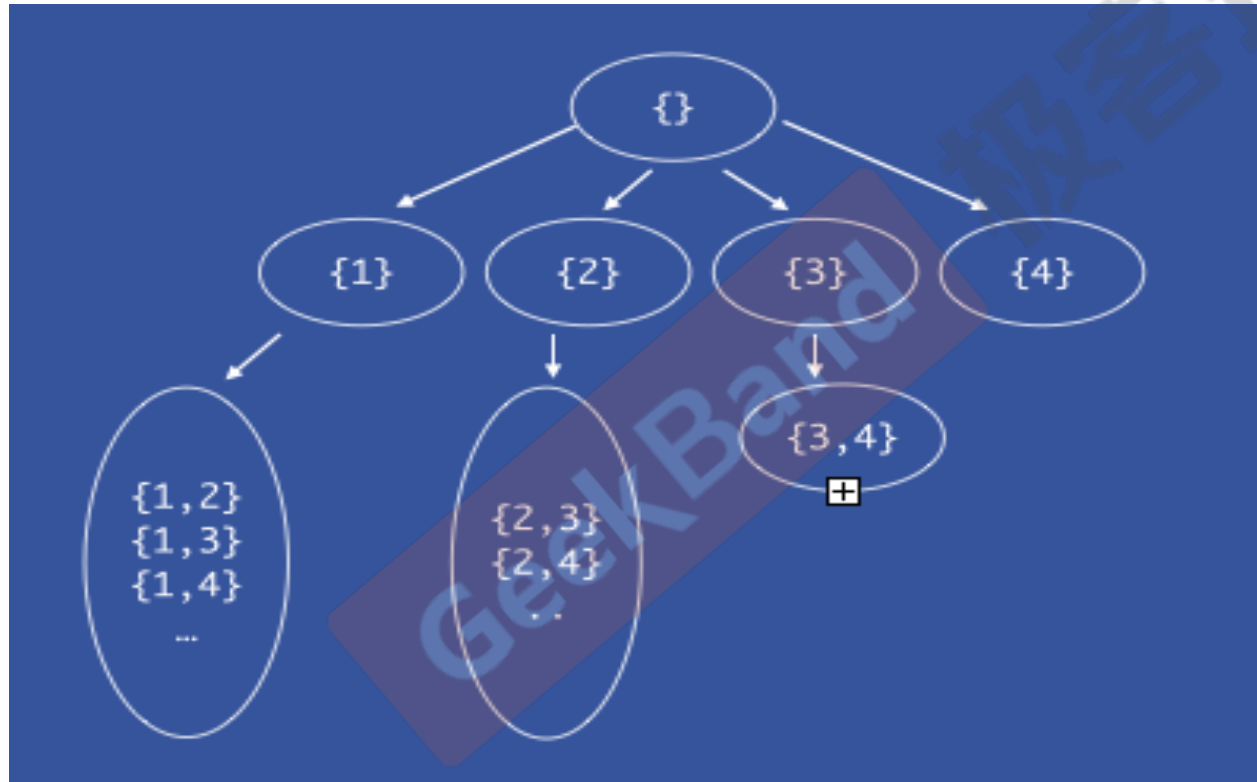
```
void subsets(int[] num) {  
    ArrayList<Integer> path = new ArrayList<Integer>();  
    Arrays.sort(num);  
    subsetsHelper(path, num, 0);  
}  
  
void subsetsHelper(ArrayList<Integer> path, int[] num, int pos) {  
    outputToResult(path);  
  
    for (int i = pos; i < num.length; i++) {  
        path.add(num[i]);  
        subsetsHelper(path, num, i + 1);  
        path.remove(path.size() - 1);  
    }  
}
```

## Subsets - Sample 1

$\{1, 2\}$



## Subsets - Sample 2



# Unique Subsets

$\{1,2,2\}$

$\{\}, \{1\}, \{2\}, \{1,2\}, \{2,2\}, \{1,2,2\}$

GeekBand

极客班



# Unique Subsets

与Subsets有关，先背下Subsets的模板

既然要求Unique的，就想办法排除掉重复的。

思考哪些情况会重复？如 $\{1, 2(1), 2(2), 2(3)\}$ ，规定 $\{1, 2(1)\}$ 和 $\{1, 2(2)\}$ 重复， $\{1, 2(1), 2(2)\}$ 和 $\{1, 2(2), 2(3)\}$ 重复。观察规律。

得出规律：我们只关心取多少个2，不关心取哪几个。

规定必须从第一个2开始连续取（作为重复集合中的代表），如必须是 $\{1, 2(1)\}$ 不能是 $\{1, 2(2)\}$

将这个逻辑转换为程序语言去判断

# Unique Subsets

```
void subsets(int[] num) {  
    ArrayList<Integer> path = new ArrayList<Integer>();  
    Arrays.sort(num);  
    subsetsHelper(path, num, 0);  
}  
  
void subsetsHelper(ArrayList<Integer> path, int[] num, int pos) {  
    outputToResult(path);  
  
    for (int i = pos; i < num.length; i++) {  
        if (i > 0 && i != pos && num[i] == num[i-1]) {  
            continue;  
        }  
        path.add(num[i]);  
        subsetsHelper(path, num, i + 1);  
        path.remove(path.size() - 1);  
    }  
}
```

# Permutations

[1, 2, 3]

[1, 2, 3], [1, 3, 2], [2, 1, 3], [2, 3, 1], [3, 1, 2], [3, 2, 1]

GeekBand

极客班

# Unique Permutations

[1, 2, 2]

[1, 2, 2], [2, 1, 2], [2, 2, 1]

GeekBand

极客班

# 排列组合模板总结

使用范围

几乎所有的搜索问题

根据具体题目要求进行改动

什么时候输出

哪些情况需要跳过

GeekBand

极客班

## 适用该模板的题目

Combination Sum

Letter Combination of a Phone Number

Palindrome Partitioning

Restore IP Address

...

GeekBand

极客班

# 工具介绍

Github

Git

Sublime

Terminal

g++/java

GeekBand

极客班

# 常见问题答疑

GeekBan 极客班



# 应届生招聘 VS 非应届生招聘

GeekBand

极客班

准备时间少？ 基础少？ 怎么刷题更高效？

GeekBan

极客班

转行找CS怎么准备?

GeekBand

极客班

面试的时候怎么表现自己(or 怎么沟通)

动态规划完全不懂怎么办

面试写出了BUG怎么办

## 推荐书籍

The Algorithm Design Manual, 作者还放出了自己上课的视频和slides - Skiena's Audio Lectures,

大部头有 Introduction to Algorithm 和 TAOCP (短期不推荐)

Cracking The Coding Interview. 著名的CTCI(又称CC150), Google, Microsoft, LinkedIn 前HR离职之后写的书, 从很全面的角度剖析了面试的各个环节和题目。除了算法数据结构等题以外, 还包含OO Design, Database, System Design, Brain Teaser等类型的题目。准备北美面试的同学一定要看。

剑指Offer。适合国内找工作的同学看看。

进军硅谷 -- 程序员面试揭秘。有差不多150题。

## 线上资源

[LeetCode Online Judge](#) - 找工作方面非常出名的一个OJ，每道题都有 discuss 页面，可以看别人分享的代码和讨论，很有参考价值，相应的题解非常多

[我的算法学习之路](#) - Lucida - Google 工程师的算法学习经验分享。