

# 19级图灵班第二次讨论课

Created By	Ⓚ Karl Han
Last Edited	@Sep 26, 2019 9:58 PM
Tags	Grade2019 TuringClass

公告内容：

具体内容：

语法、概念

二分查找思想与递归迭代实现

讲课内容随想(可能不讲的)

## 公告内容：

时间：本周六(2019.9.28)，上午九点

地点：计算机学院131

主题：19级图灵班第二次讨论课。

主要内容：函数、递归、指针、动态数组。

## 具体内容：

### 语法、概念



- 函数是什么？
  - 本质上就是实现某种功能的一个过程，使用者只需要知道函数功能和形式，便能从中实现某种功能。
  - 黑盒子，使用者无需知道具体实现
  - 抽象过程的一种体现
  - 就像 `gen_rand_arr(num)` 一样，单纯实现某种功能。
  - 为什么不要使用一个函数实现？如main？
    - 调试难度

- 人脑记忆
- 代码复用
- 递归：自己调用自己的过程
  - 例子：斐波那契数列的定义，然后试一下进行定义程序
  - 递归寻找最大最小值（Divide and Conquer）
    - 我们需要找的最大值就是比较当前值和除了当前以外的剩余数组的最大值的较大值
    - 当我们找到只有一个元素的数组，那么我们就可以很容易获得当前数组的最大值
    - 同理较小值
- 指针：存储的是指定类型变量的地址

```
int num = 123;
// 这是一个指向int类型变量的指针
int *a;
// 让变量a存储int类型变量num的地址
a = &num;
```

- 介绍十六进制
- 指针大小固定，存储地址
- 从内存上看，本质上所有都是通过二进制串来表示信息，区别在于如何进行解析
- 对比数组和指针的不同
  - 数组可以直接通过 `sizeof` 获取占用字节数
  - 指针可以动态分配空间，但是使用完了需要手动free
- 二级指针
  - 数组形式arr[m][n]，固定大小，占用空间m \* n
  - 二级指针首先指向指针数组（e.g.元素类型都是int\*），再分配每个元素大小
  - 动态分配空间更加灵活，数组整个需要连续的空间，而动态分配不需要

## 二分查找思想与递归迭代实现

▼ Thanks to 润智，代码来自去年讲课，有修改

▼ Reference: [https://en.wikipedia.org/wiki/Binary\\_search\\_algorithm](https://en.wikipedia.org/wiki/Binary_search_algorithm)

## 讲课内容随想(可能不讲的)

- 我们为什么要函数？
  - 抽象：实现某种功能，只需要提供某种参数，则得到某种形式的输出
- 类型转换
- 介绍C++里面的类和对象
- 对比Rust和C++两种语言，让同学们对不同编程语言的区别有所了解（链表）