拉勾Offer工场

主讲人：ccmouse

2017-09-23

ccmouse

自由职业者，赛车手

原谷歌高级软件工程师

谷歌，摩根士丹利校招命题组成员


上海工程技術大學


TRACKDAY
by J.DONG
10


BRIDE
OMP


HAWK
PERFORMANCE
2017上海市精英赛
DELINTE
TIANMA

1

CHAPTER1

编码能力

2

CHAPTER2

算法和数据结构

3

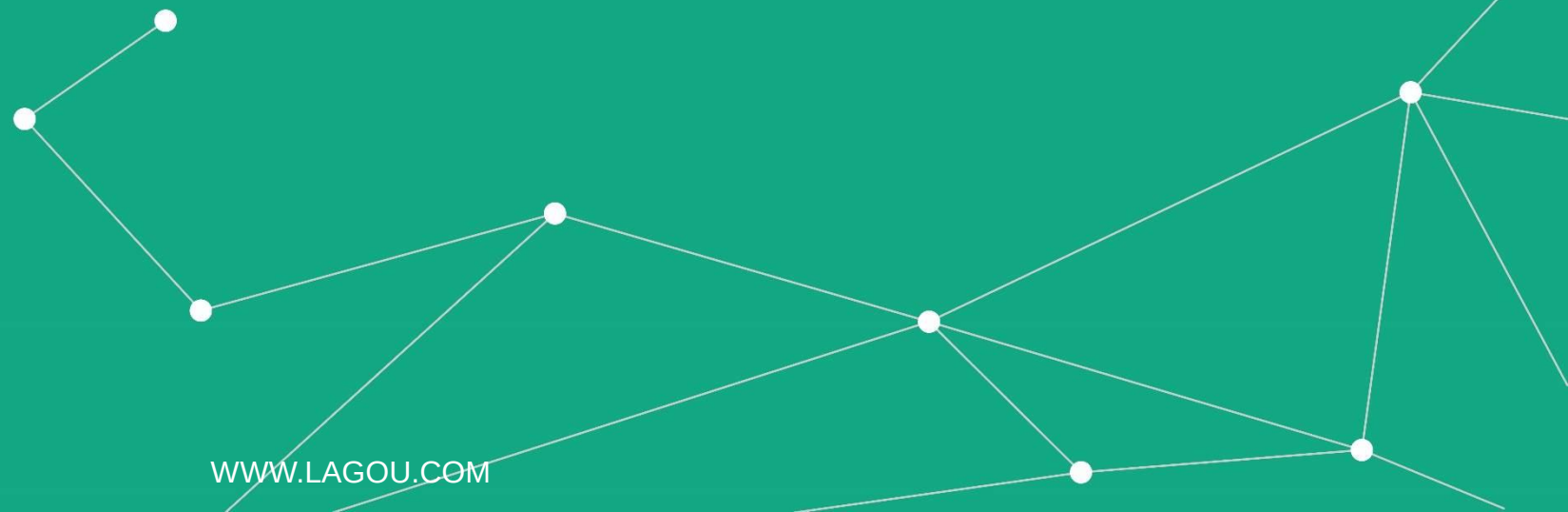
CHAPTER3

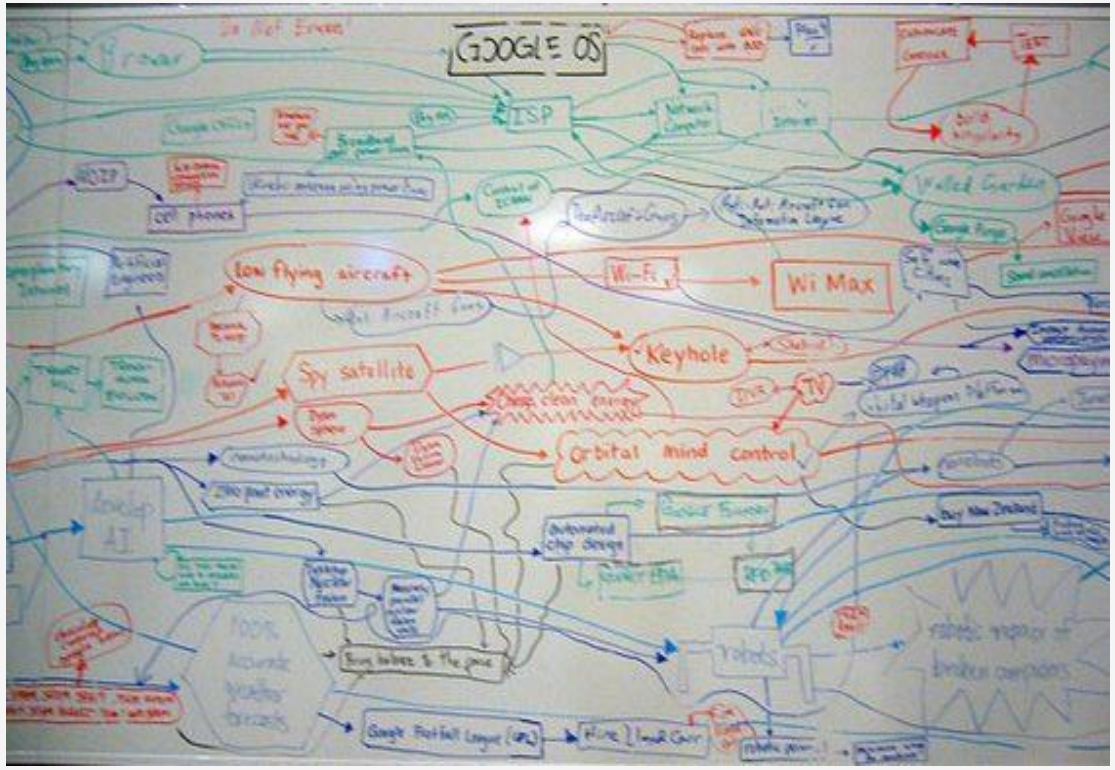
基础知识

CHAPTER1

编码能力

Talk is cheap. Show me the code.




GOOGLE OS
ISP
Low flying aircraft
Wi-Fi
Wi Max
Keyhole
Spy satellite
Orbital mind control
Google Ratball League

在白板上写程序

程序写在：白板，纸笔，Word文档，记事本。。。

修改不便；缩进不便；对齐困难

心里不抵触；先思考后写；不要惧怕修改/重写

数学归纳法回顾

用于证明断言对所有自然数成立

1. 证明对于N=1成立
2. 证明N>1时：如果对于N-1成立，那么对于N成立

数学归纳法的正确性

公理

递归控制

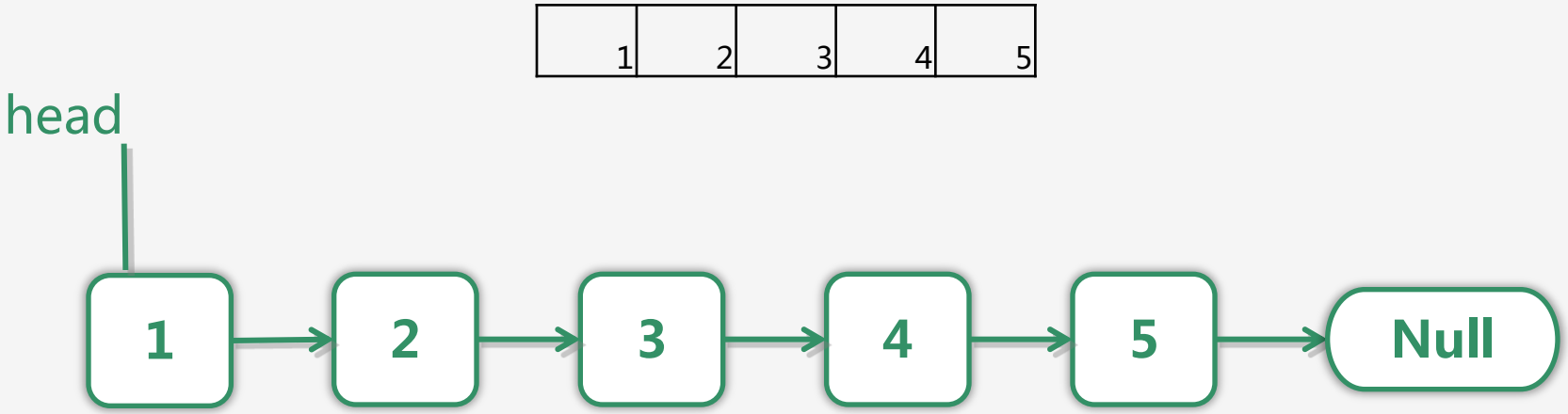
编码：数学归纳法中的数学/自然语言 → 程序语言

正确性证明：程序语言 → 数学归纳法中的数学/自然语言

递归的写法

- 严格定义递归函数作用，包括参数，返回值，Side-effect
- 先一般，后特殊
- 每次调用 必须 缩小问题规模
- 每次问题规模缩小程度必须为 **1**

例一：链表创建

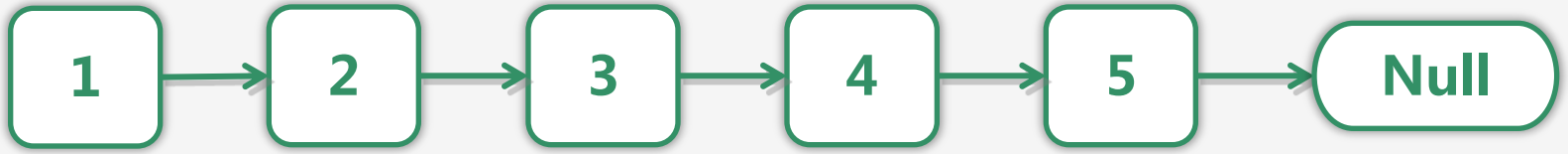


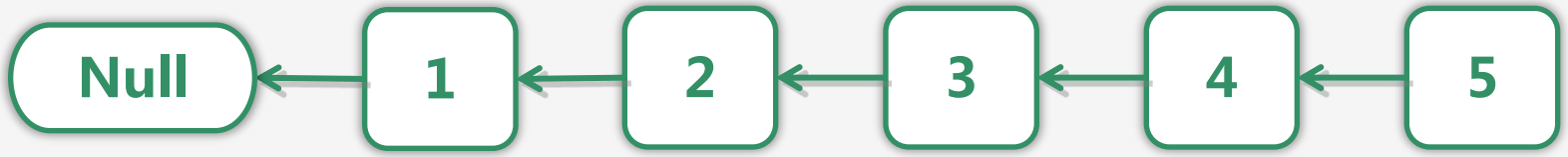
例一：链表创建

Node CreateLinkedList(List<Integer> values)

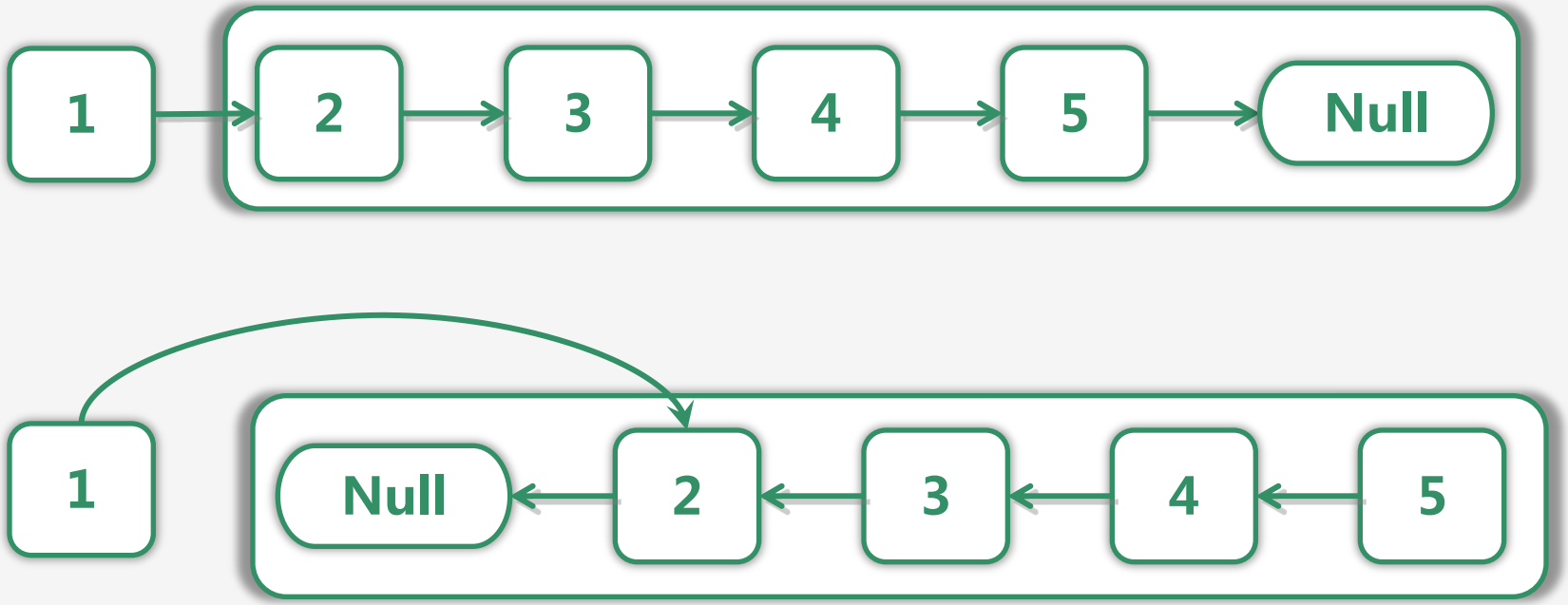
创建一个以values.get(0)开头（如果有的话），以Null结尾的链表
返回其头指针

例二：链表反转





例二：链表反转



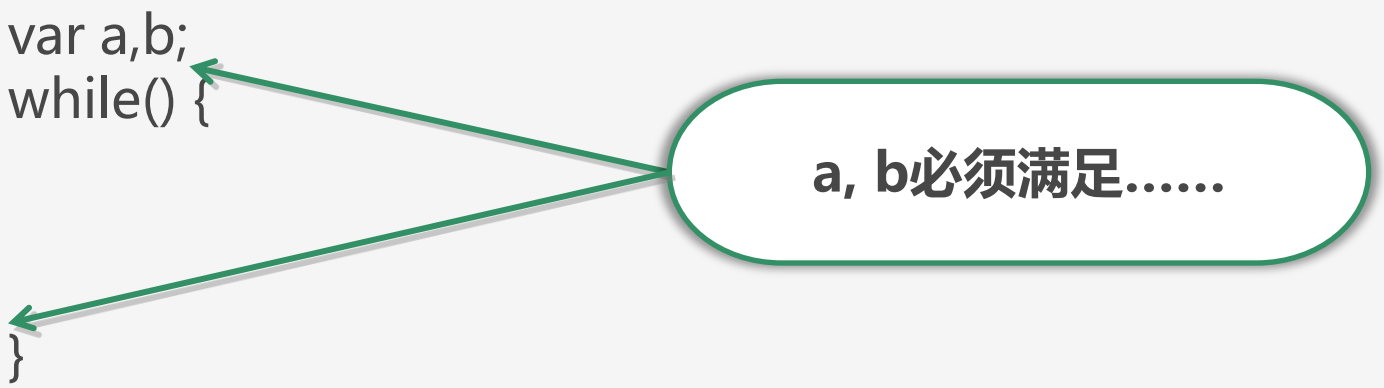
递归的缺点

StackOverflow!

循环控制

不要尝试递归 → 非递归

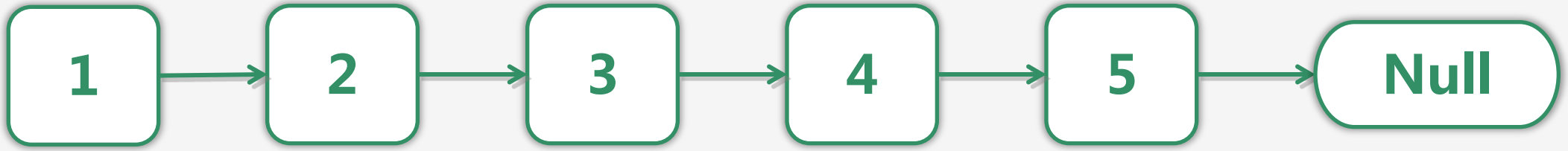
循环不变式：Loop Invariant

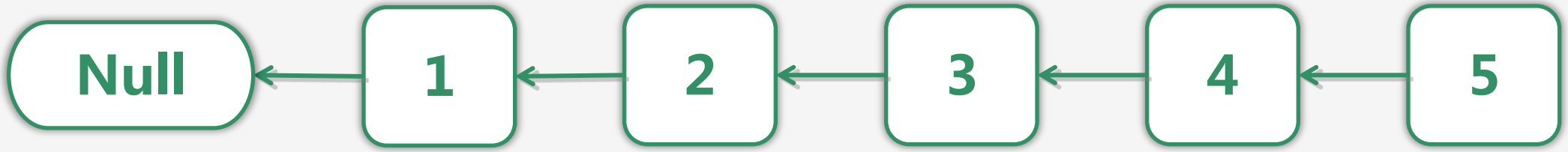


循环的写法

- 定义循环不变式，并在循环体每次结束后 保持 循环不变式
- 先一般，后特殊
- 每次 必须 向前推进循环不变式中涉及的变量值
- 每次推进的规模必须为 **1**

例一：链表反转





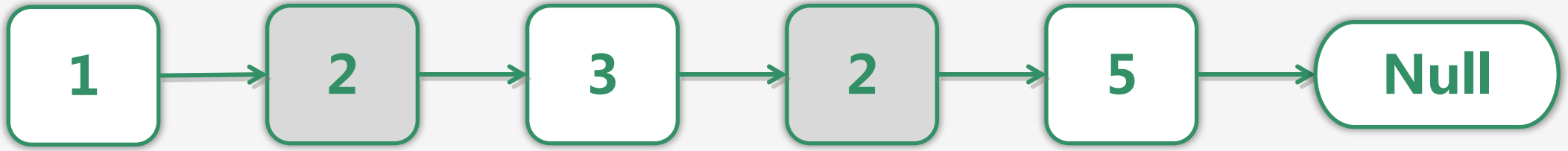
例一：链表反转

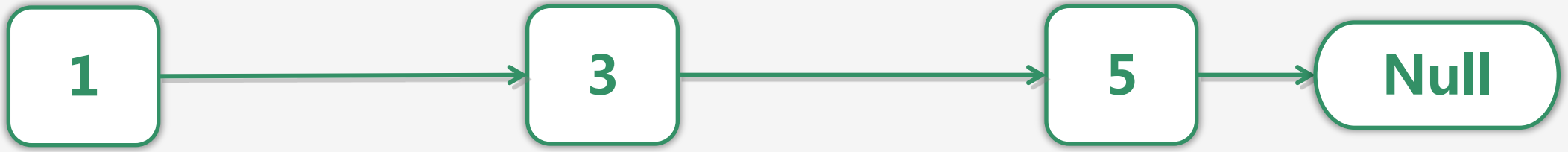
循环不变式：

currentHead 指向待反转链表头

newHead 指向已反转立案表头

例二：链表中delete_if





例二：链表中delete_if

循环不变式：

prev 不空，并且从头结点至prev结点的所有
满足需要删除的结点已经删除

边界控制

例：二分查找

- 在有序数组中查找元素k，返回k所在下标
- binarySearch([1, 2, 10, 15, 100], 15) == 3

边界控制

二分查找思路

- 规定要查找的值k可能在的数组arr内下标区间a, b
- 计算区间a, b的中间点m
- 若k<arr[m]，将区间缩小为a, m，继续二分查找
- 若k>arr[m]，将区间缩小为m, b，继续二分查找
- 若k==arr[m]，则找到元素位于位置m

边界控制

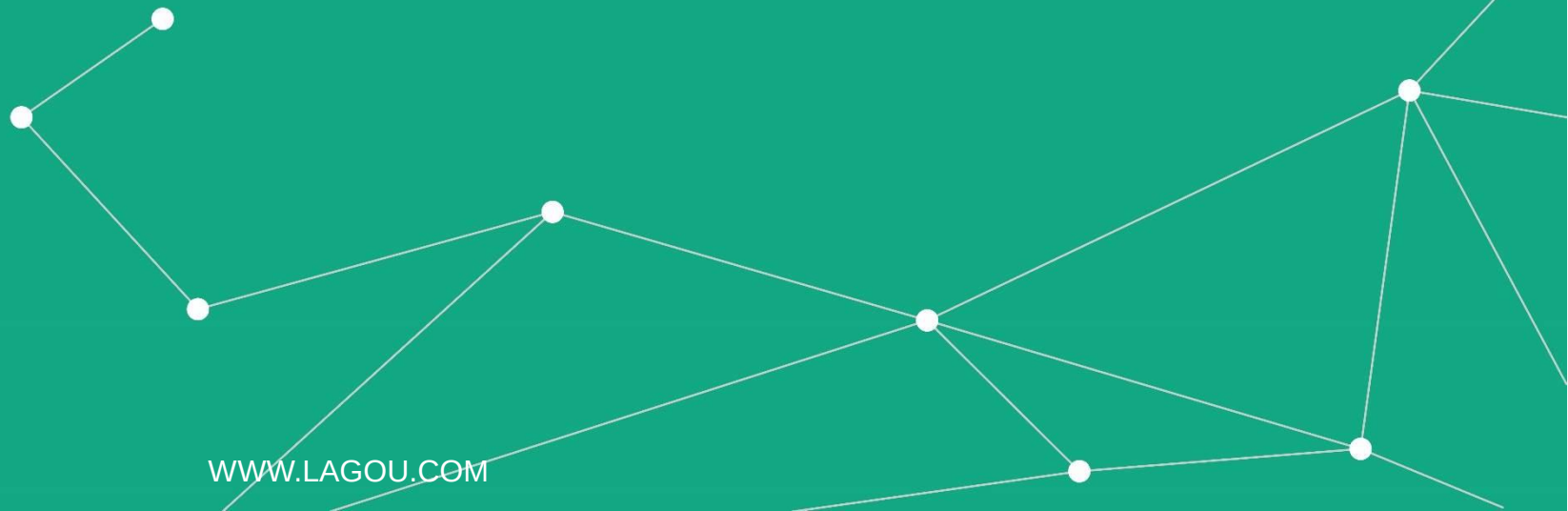
例：二分查找

- 程序员很难把二分查找写正确
- 善用特殊值检查边界处理

CHAPTER2

算法和数据结构

树，图，复杂度



算法与数据结构

- 贪心
- 树的遍历
- 深度优先，广度优先

贪心

例：区间合并

https://leetcode.com/problems/merge-intervals/description/

贪心

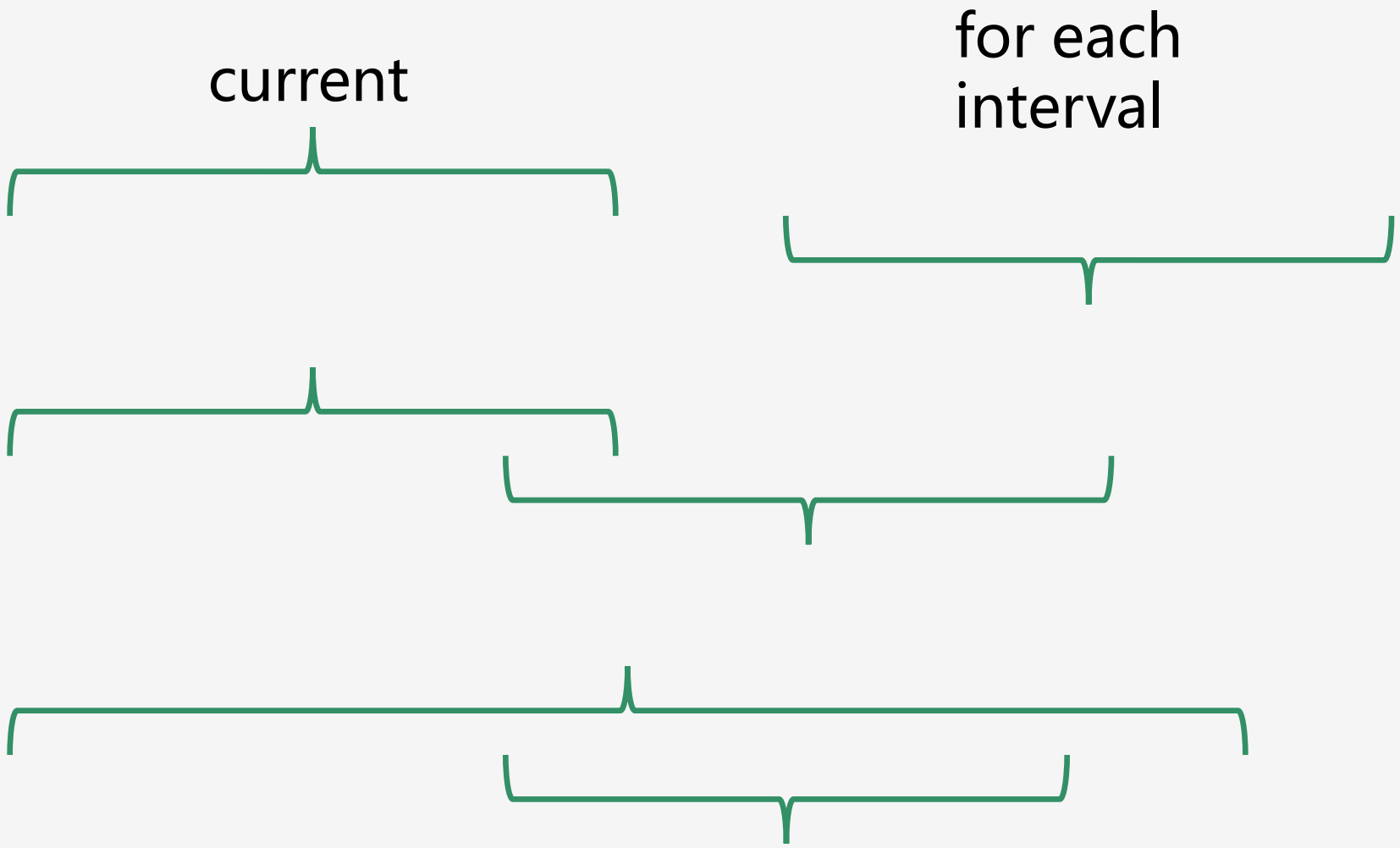
已合并完成的区间

正在合并过程中的区间: current

下一个待合并区间

未处理区间

例：区间合并

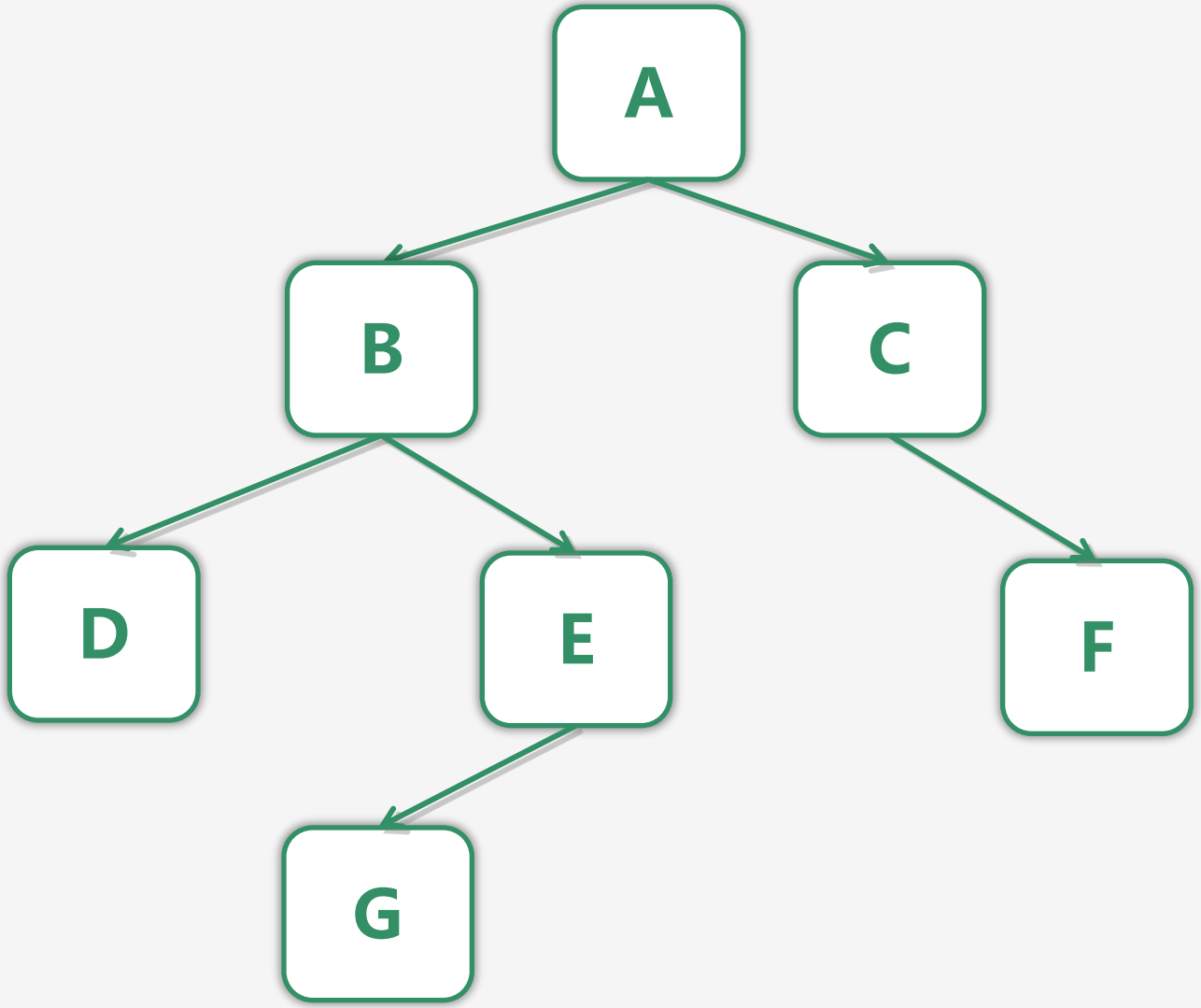


二叉树的遍历

如何进行前序遍历

- 先遍历树根
- 然后前序遍历左子树
- 再前序遍历右子树

二叉树的遍历



前序：ABDEGCF

中序：DBGEACF

后序？

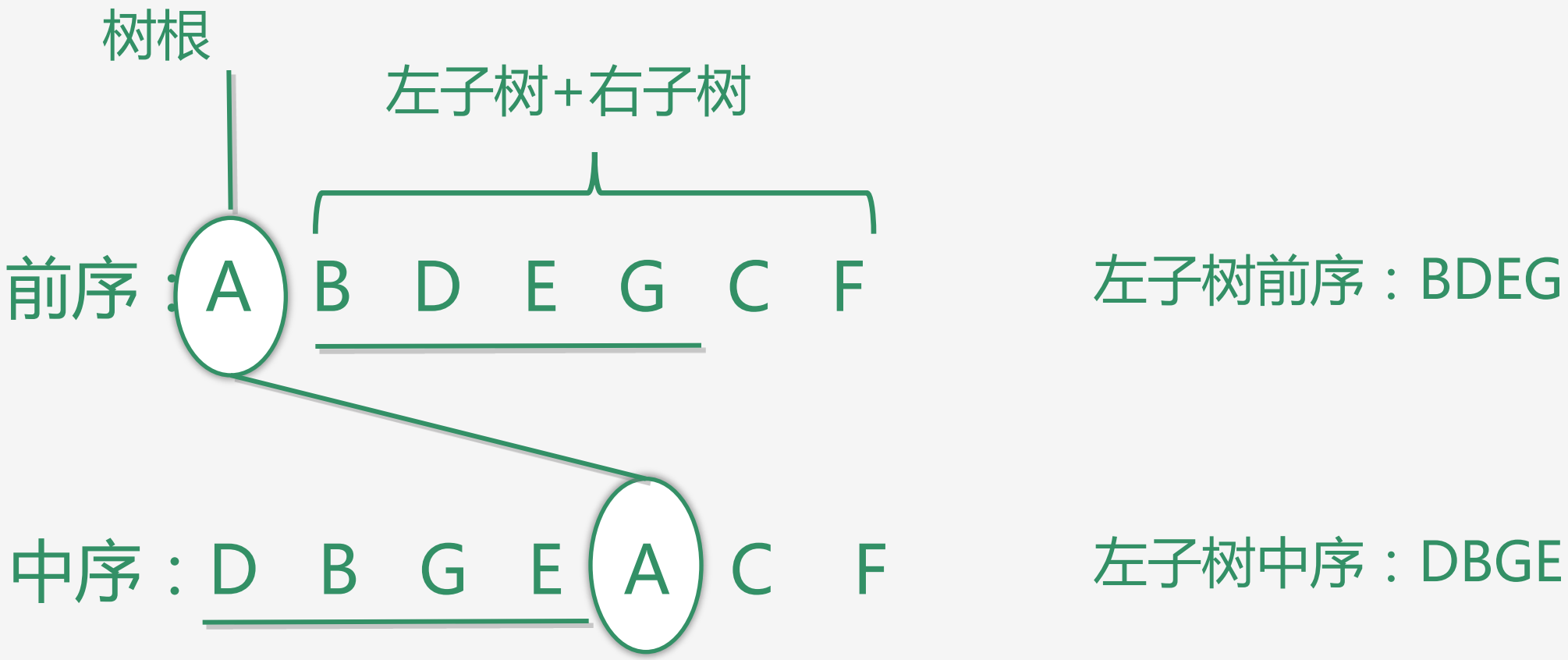
例：根据前序中序构造二叉树

前序：ABDEGCF

中序：DBGEACF

后序？

例：根据前序中序构造二叉树



深度优先，广度优先

例：走迷宫

算法复杂度

O(N!), O(2^N), O(N^2), O(NlogN), O(N), O(logN)……

- 代表最坏情况用时

$$f(x) = O(g(x)) \text{ as } x \to \infty$$ 当且仅当

$$|f(x)| \leq M|g(x)| \text{ for all } x \geq x_0$$

算法复杂度

O(N!), O(2^N), O(N^2), O(NlogN), O(N), O(logN)……

- 10^8 ~ 秒级
- 最大的N分别大约是：
- 10, 30, 10000, 10^7, 10^8, 天文数字

算法复杂度

O(N^2)

```
for (int i = 0; i < n; i++) {
        for (int j = i; j < n; j++) {
            ...
            ...
        }
}
```

- 插入排序，选择排序

算法复杂度

O(NlogN)

f([……]) → f([…]) + f([…])，花O(N)时间拆分

- 归并排序，快速排序（平均）

算法复杂度

O(logN)

f([……]) → f([…]) or f([…])，花O(1)时间拆分

- 二分查找

算法复杂度

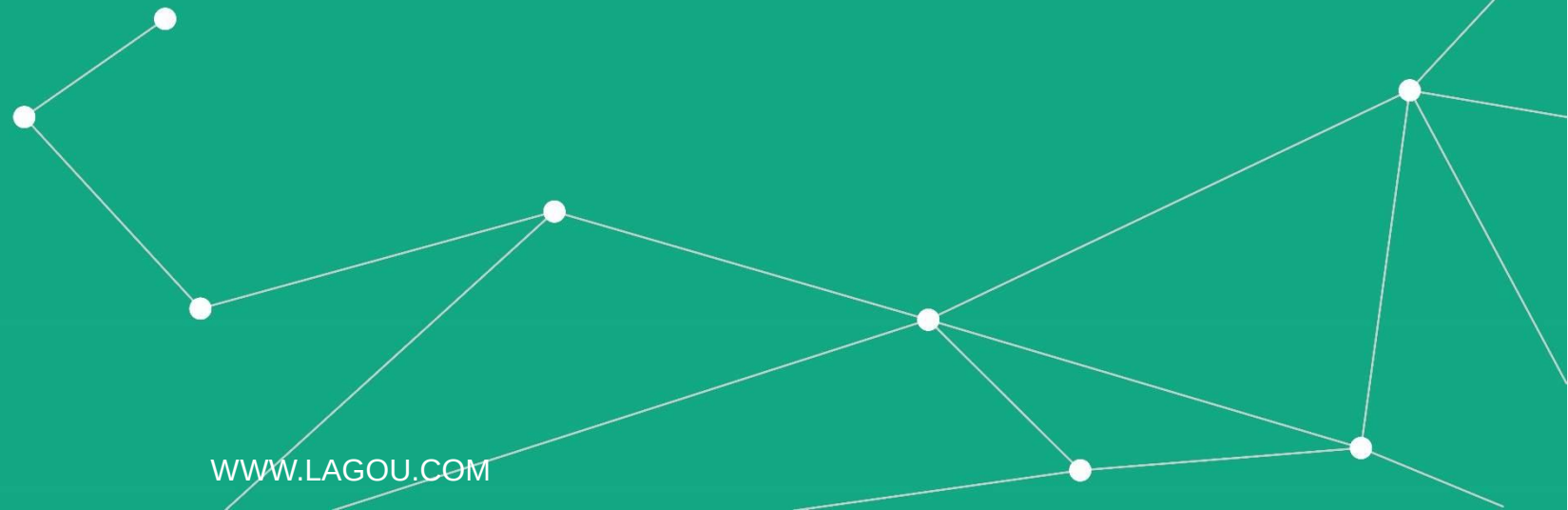
算法的组合　　例：区间合并

- 排序　　O(NlogN)
- 扫描已排序的列表　　O(N)
- 总复杂度？
- O(NlogN)

CHAPTER3

基础知识

操作系统，网络，服务器编程



操作系统

- 进程 vs 线程
- 寻址
- 进程间通信

进程

- 包含线程，内存，文件/网络句柄
- 拥有独立地址空间

线程

- 包含栈，PC(Program Counter)，TLS(Thread Local Storage)
- 同一进程的线程共享地址空间

寻址 int a = *p

1. 指针p指向逻辑地址
2. 映射p到物理内存中的地址
3. 若p不在物理内存中，把p所在的分页装入物理内存
4. 若物理内存不足，需要选择一个分页交换出去
5. 从p所在物理内存读取数据装入寄存器

数据类型相关

- 整数类型
- 浮点数
- Boxing and Unboxing

整数类型

32位int的范围？

- -2^31 ~ 2^31 - 1
- 但是，直观的方法？

整数类型

直观的方法

31位

- +111……111 → 2^31 – 1
- -111……111 → -(2^31 – 1)
- +0 ? -0?

整数类型

补码

- 取反加1
- 例：-1
- 0000...1 → 取反 → 1111...0
- 1111...0 → +1 → 1111...1

整数类型

补码的优点

- -1 + 1 = 0
- 1111...1 + 0000...1 = 0000...0
- 唯一表示0，没有+0, -0
- 总共表示2^32个数

浮点数与定点数

浮点数　(+/-)1.xxx * 2^y

- 符号位 | 指数部分 | 基数部分
- 64 位 double 范围：+/- 10^308
- 64 位 double 精度：10^15

浮点数与定点数

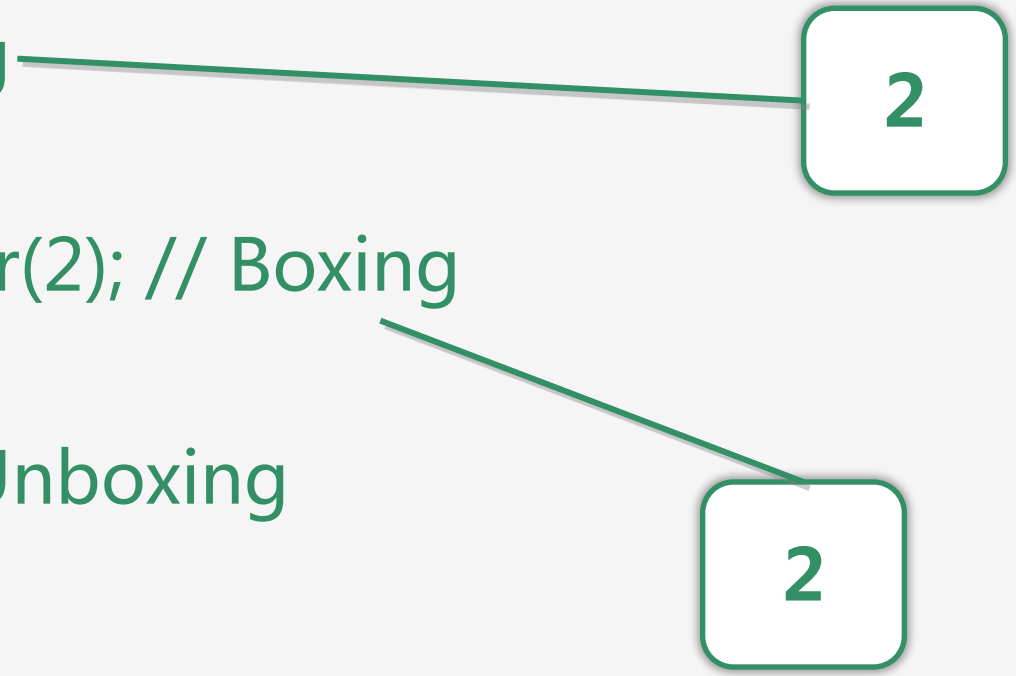
浮点数比较

- a == b ?
- Math.abs(a – b) < eps ?
- 使用 BigDecimal 算钱

Java 数据类型

Boxing and Unboxing

- Integer a = 2; // Boxing
- Integer b = new Integer(2); // Boxing
- int v = a.intValue(); // Unboxing



Java 数据类型

Boxing and Unboxing

- new Integer(2) == 2 ?
- new Integer(2) == new Integer(2) ?
- Integer.valueOf(2) == Integer.valueOf(2) ?
- Integer.valueOf(2).intValue() == 2 ?
- new Integer(2).equals(new Integer(2)) ?

进程间通信

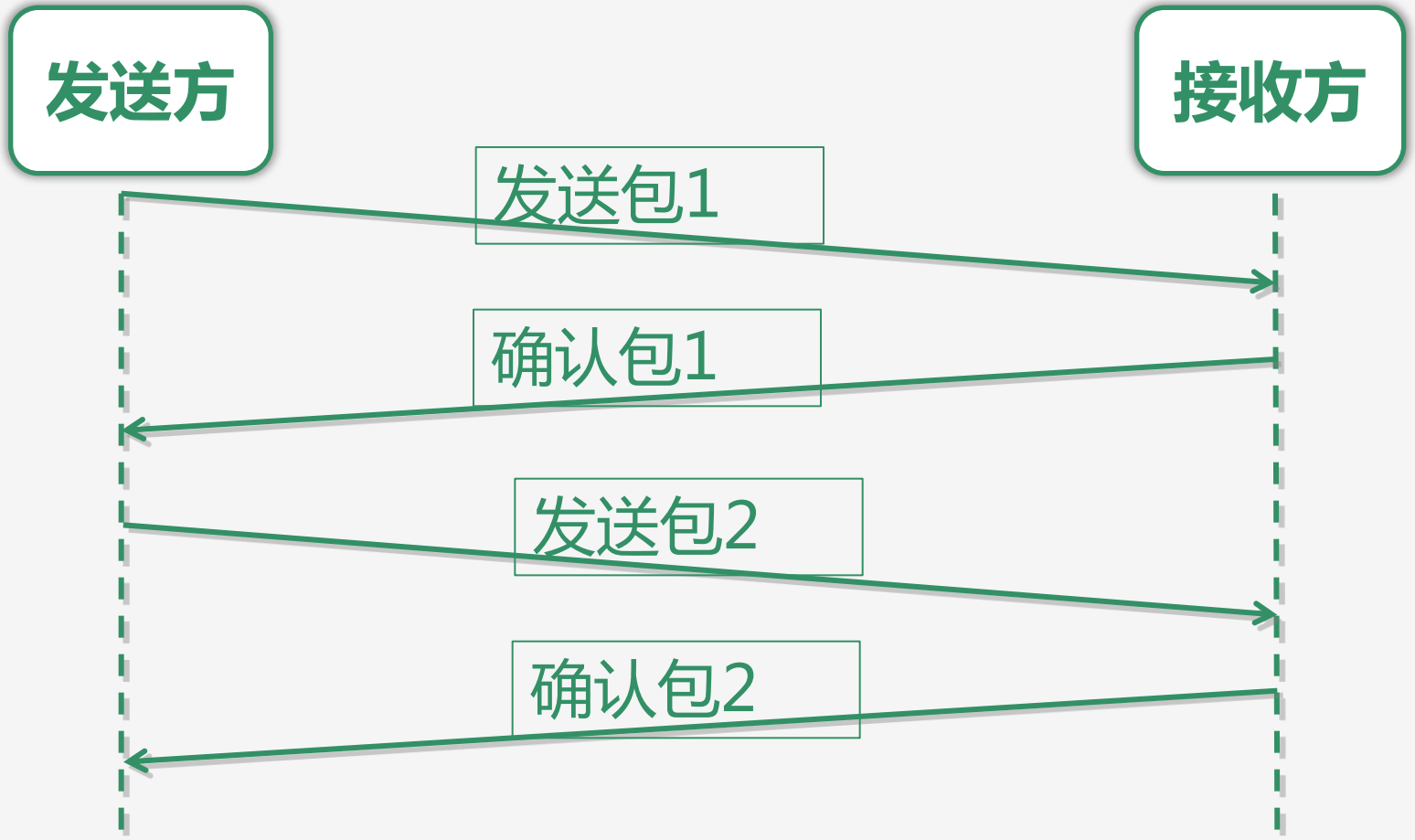
- 文件
- 管道/命名管道
- Signal
- 网络

网络

- TCP协议 - 滑动窗口 + 抓包演示
- 服务器编程

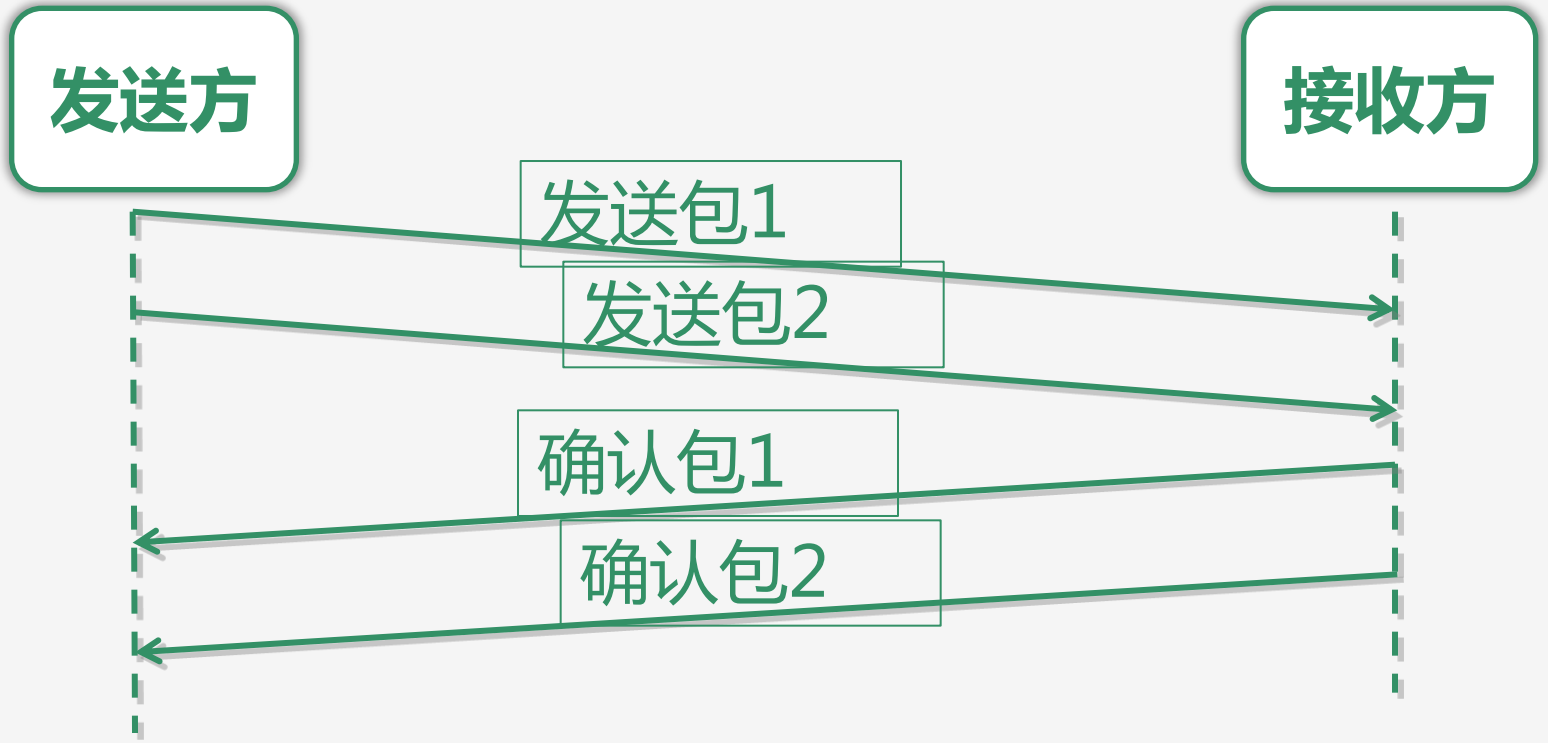
滑动窗口

问题提出

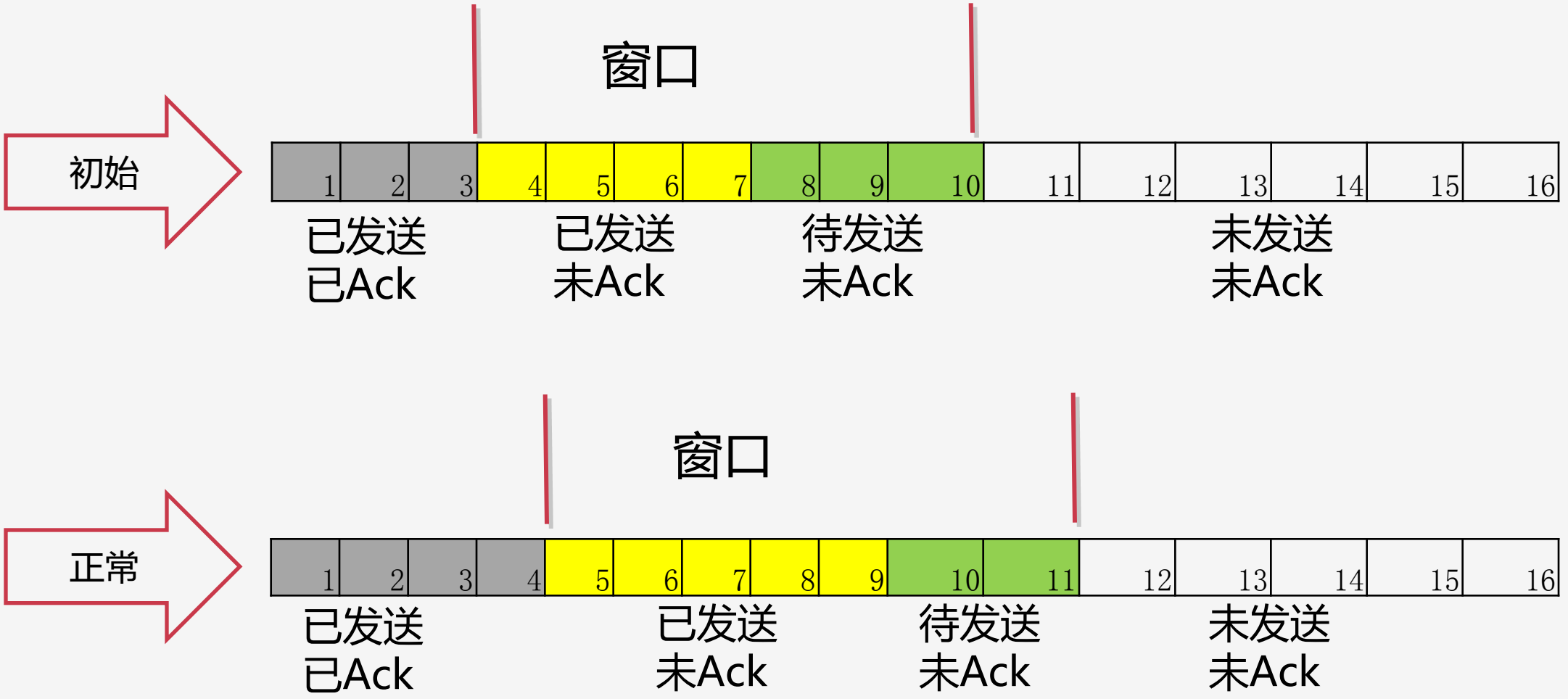


滑动窗口

改进方案

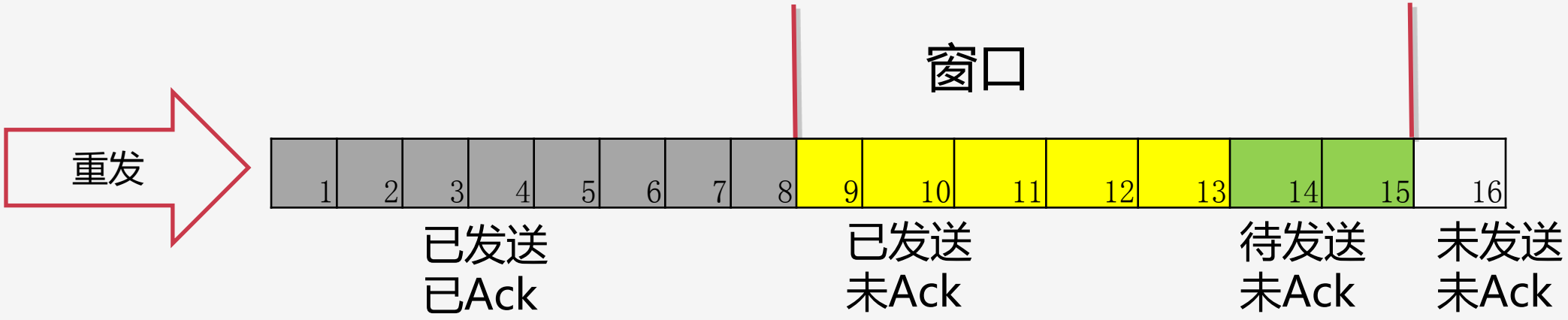


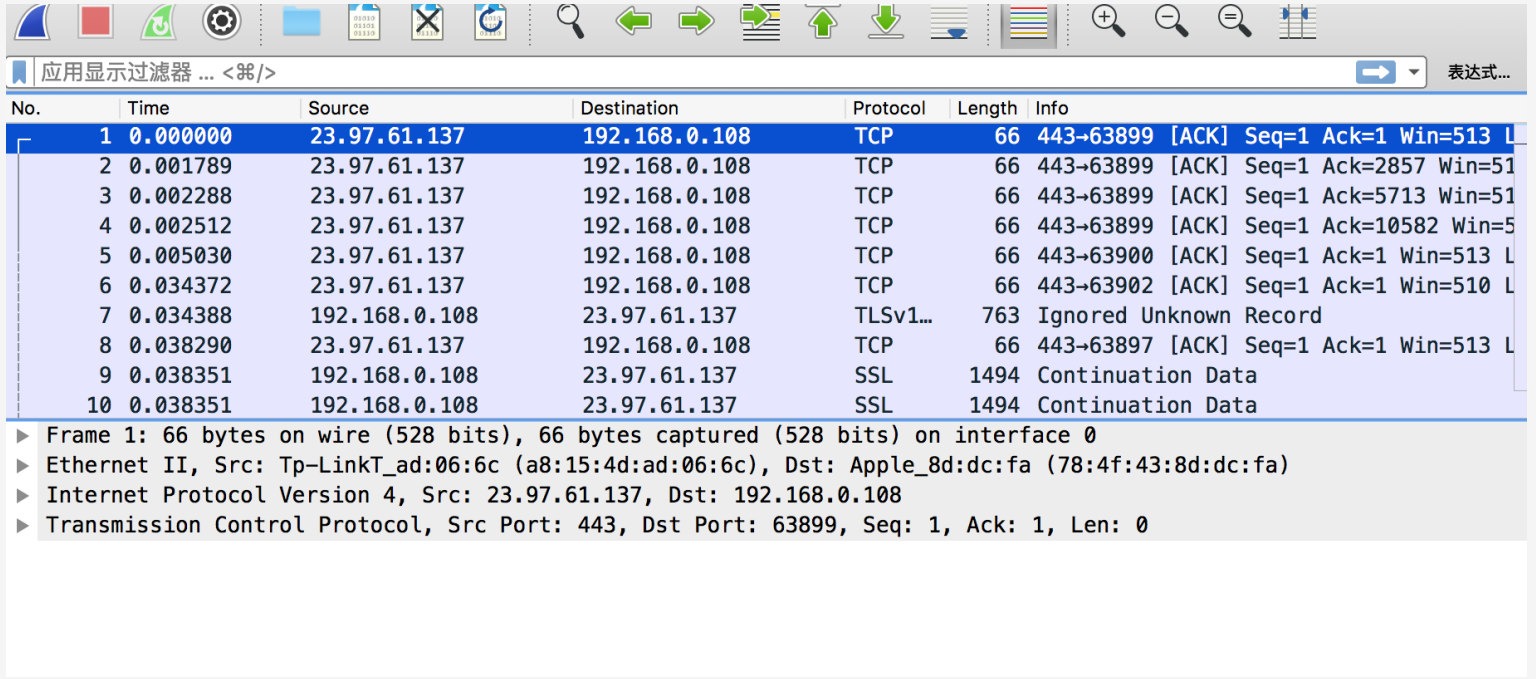
滑动窗口实现


初始
窗口
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
已发送
已Ack
已发送
未Ack
待发送
未Ack
未发送
未Ack
正常
窗口
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
已发送
已Ack
已发送
未Ack
待发送
未Ack
未发送
未Ack

滑动窗口实现


窗口
丢Ack
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
已发送
已Ack
已发送
未Ack
未发送
未Ack


窗口
重发
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
已发送
已Ack
已发送
未Ack
待发送
未Ack
未发送
未Ack



抓包分析：浏览器打了网址后发生了什么

服务器编程

- 单线程
- 线程池
- 异步IO

添加拉勾课程咨询老师薇薇微信，获得更多课程信息；

关注互联网offer之路，获取海量互联网求职干货。

Thank You