作业问题

- 基础作业 错误较多我会讲解一下
- 基础作业 阅后删除
- 测试作业 每天请务必完成

小组

- 自习的时候打开zoom
- 有问题找小组 不明确的问题发qq群
- 小组每天要交流笔记

代码问题

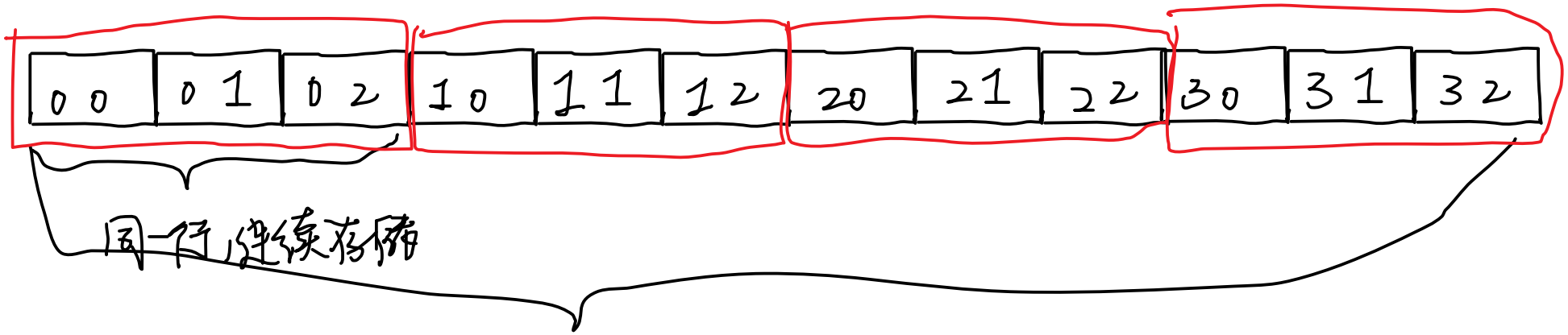
- 拿出纸和笔，分析一下问题，提出解决方案，然后检查并调整
- 锻炼自己的调试能力 定位问题 监视所有的变量
- 找到问题 不知道怎么解决，提问

今天的安排

- 播放龙哥 30min
- 复习 指针和数组
- 我讲解二级指针和函数指针
- 讲解习题 103个数

二维数组的内存分布

int arr[4][3]



行与行之间连续存储.

如何理解数组的数组?

① 将二维看成一维 ⇒ 大小为4

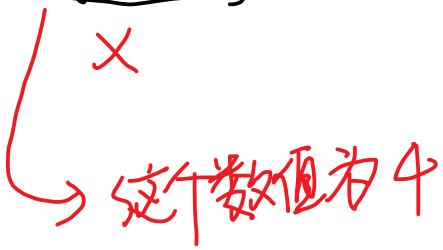
② 每一个元素又是一个数组 ⇒ 大小为3

大小问题

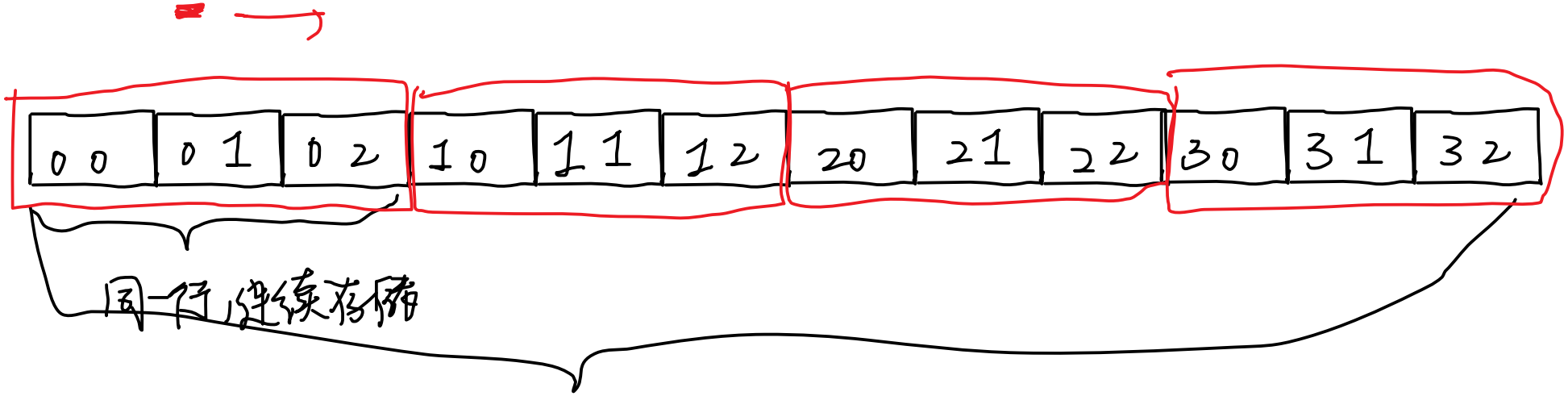
使用 malloc memset.

① char str[30]; 如何使用 memset，清空 str.

② char str[30]; func(str) { memset(str, 0, sizeof(str)) }



二维数组的地址计算



行与行之间连续存储.

int arr[4][3]

arr+1 1*sizeof(int)*3

(int[3])*

*arr+1 1*sizeof(int)

int[3] → int*

arr[1]+1

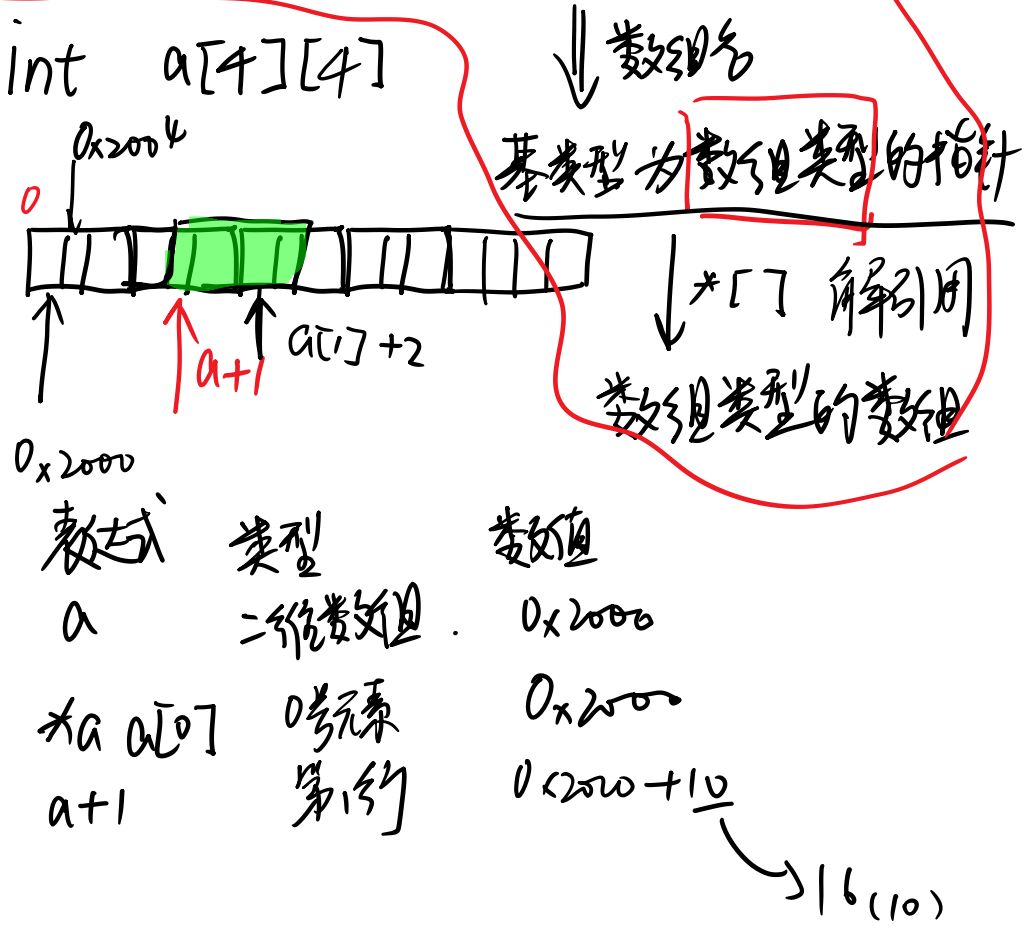
↓

*(arr+1)+1

复习数组指针

表 示 形 式	含义	地址值
a	二维数组名，指向一维数组 a[0]，即 0 行首地址	0x2000
a[0]， *(a+0)， *a	0 行 0 列元素地址	0x2000
a+1，&a [1]	1 行首地址	0x2010
a [1]，*(a+1)	1 行 0 列元素 a[1][0] 的地址	0x2010
a[1]+2， *(a+1)+2， &a[1][2]	1 行 2 列元素 a[1][2] 的地址	0x2018
*(a[1]+2)， *(*(a+1)+2)， a[1][2]	1 行 2 列元素 a [1] [2] 的值	元素值为 13

⇒ 二维数组 ⇒ 元素为数组的一维数组



常见问题

```
int a[4][4];
int *p = a;
int *q = a[0];
```

⇒ p q 的数值 & 数据类型 都一样

能够说明 a 和 a[0] 一样吗？

行比是绝对的。

关注编译警告

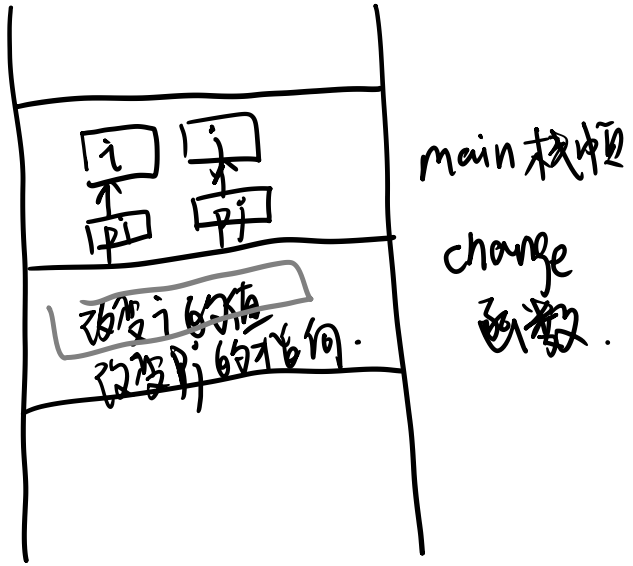
= 初始化
= 赋值

左边 = 右边

把右边的 数值，拷贝到左边所在的内存

左边的数据类型 由定义决定的。

二级指针

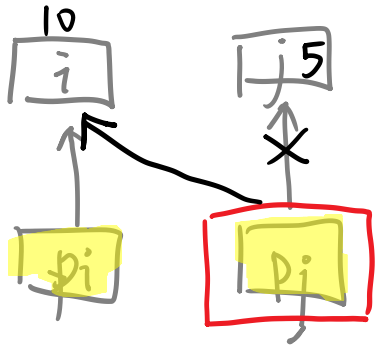


① 改变i的值 传入i的地址.

②

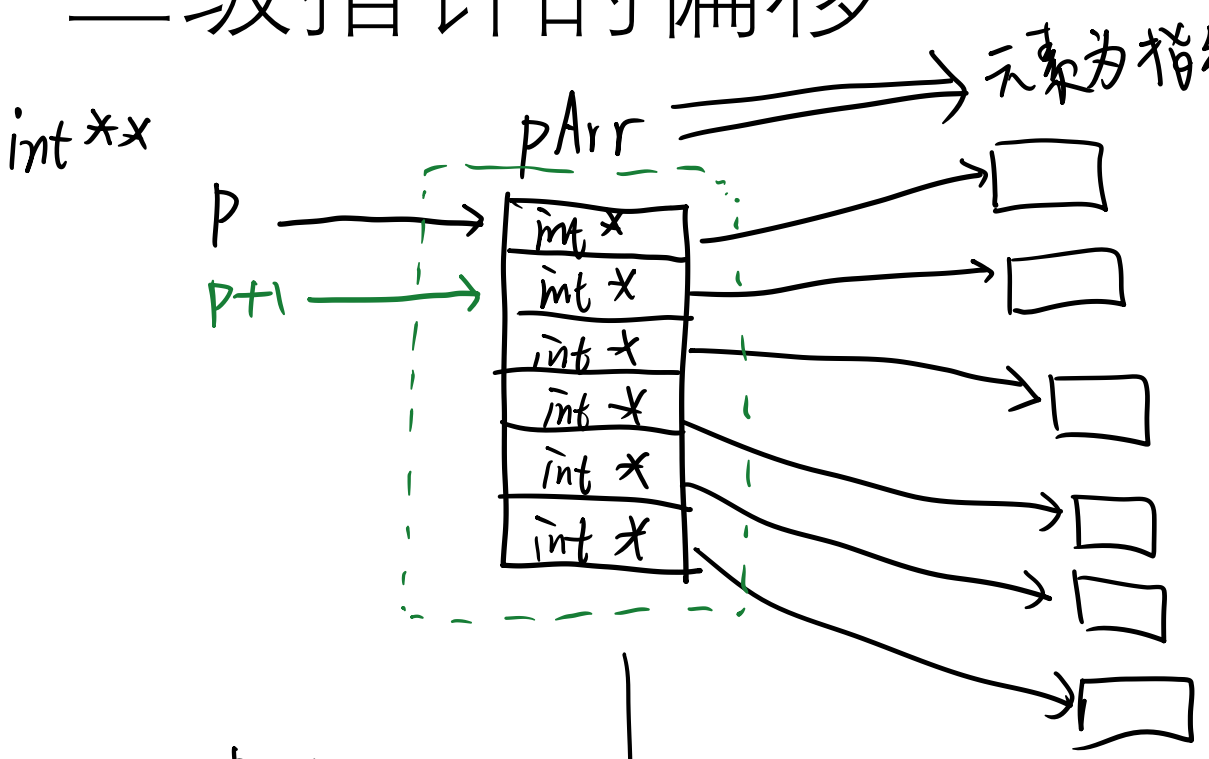
```
void change(int **p, int *pi) {
    *p = pi;
}
```

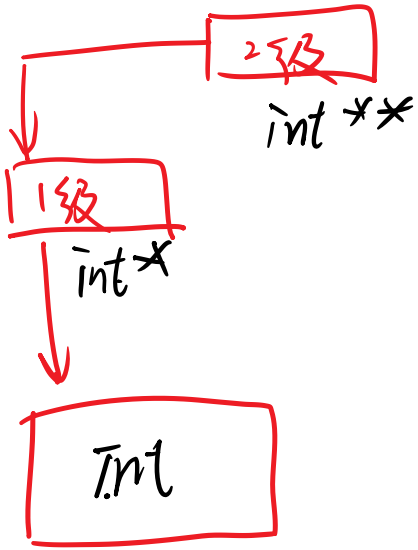
```
change(&pj,pi);
```



什么是改变指针的指向. ⟹ 修改指针变量存储的值

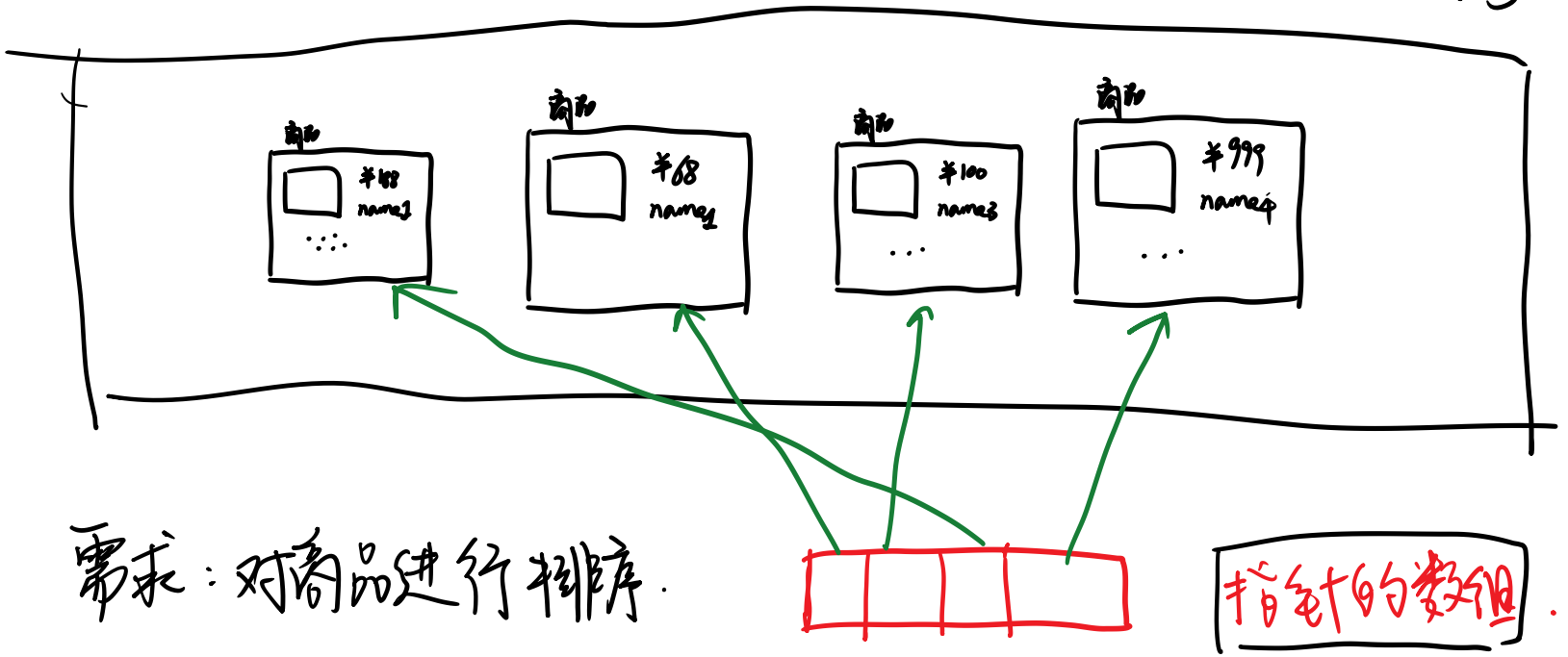
二级指针的偏移





使用二级指针的偏移一定要有指针构成的数组

指针数组的实际场景



需求：对商品进行排序.

指针的数组.

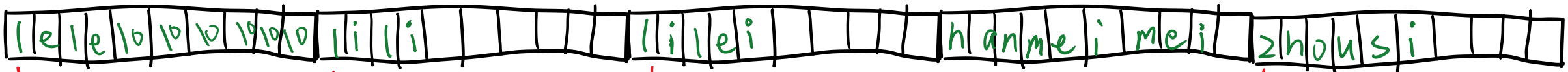
索引式排序

使用指针数组排序字符串

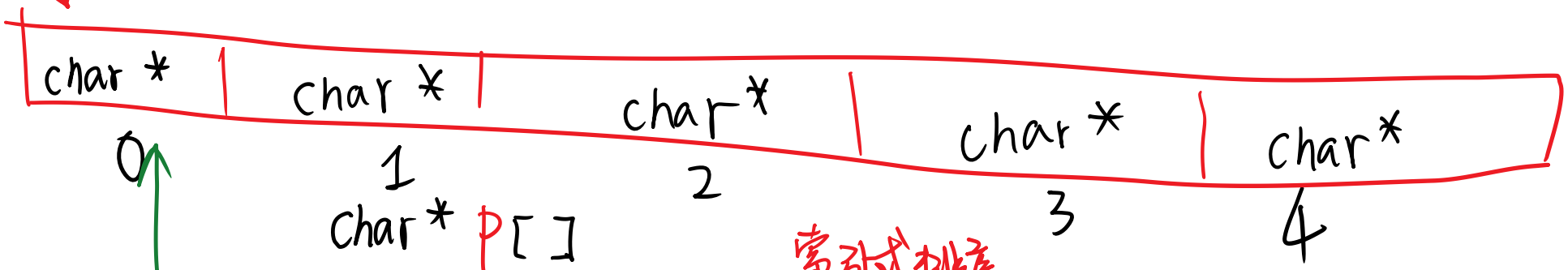
回顾二维数组

```
char b[5][10]={"lele","lili","lilei","hanmeimei","zhousi"};
```

b → char[10] *　　b[0] → char[10]/char *



char *

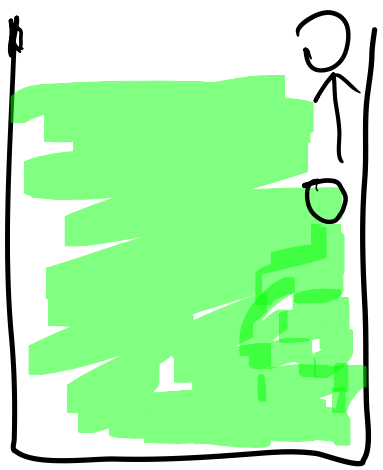


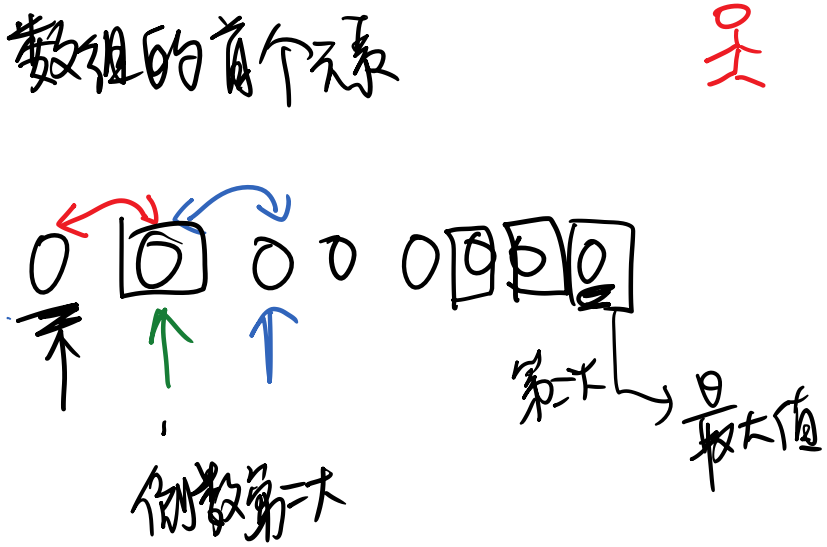
char * p[]

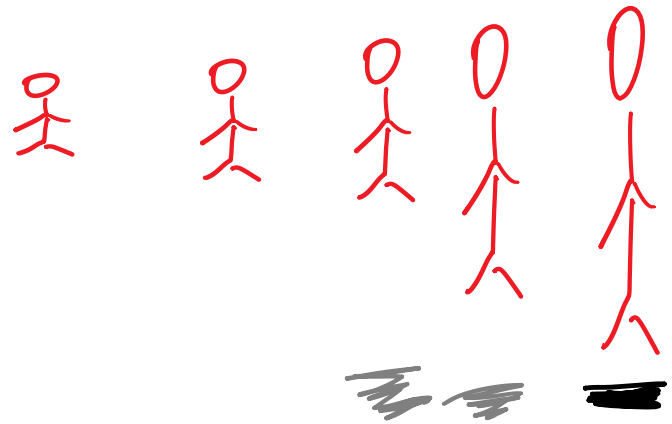
p2 · char**

索引式排序
①速度快
②不修改原数据

冒泡排序法

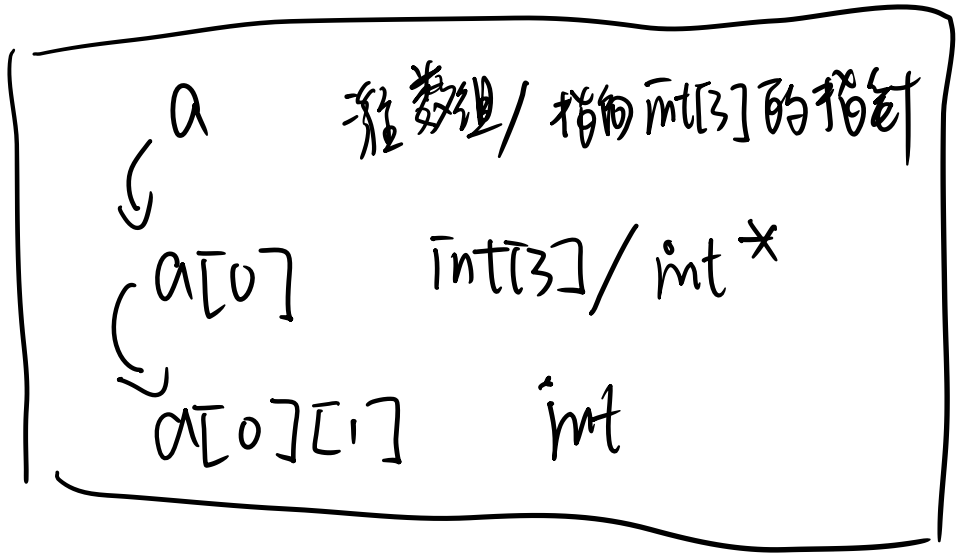






二级指针vs二维数组 int a[4][3]

在默认情况下，两者没有联系



二维数组的解引用流程

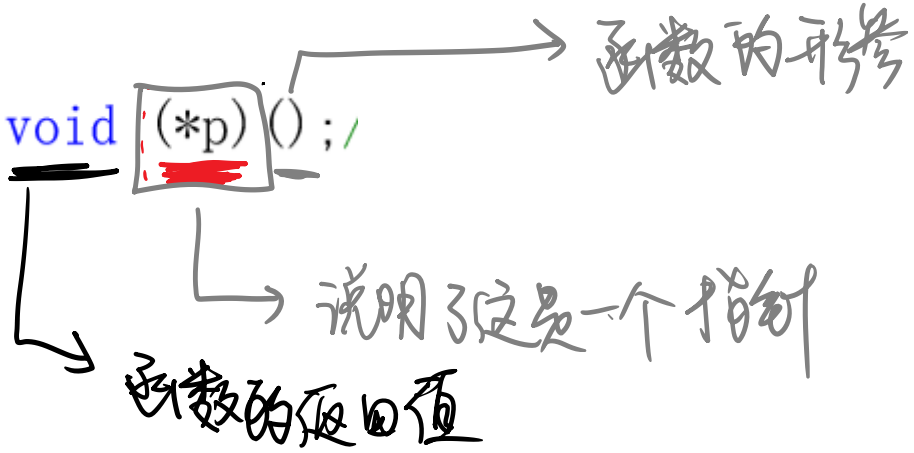
不涉及二级指针

函数指针

新的指针类型，存储函数的入口地址

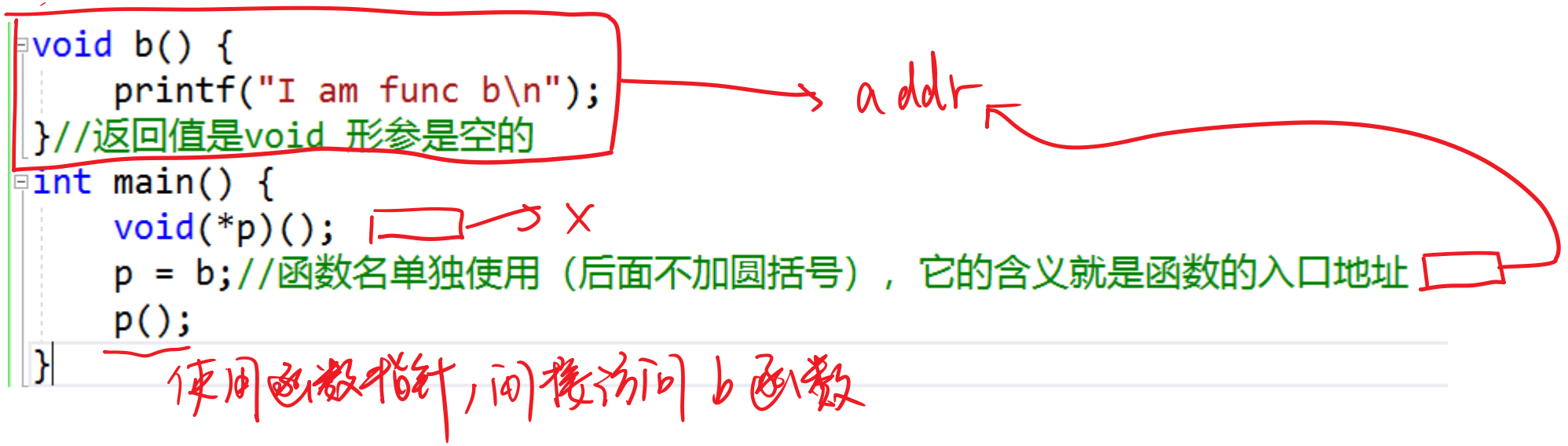
函数指针的功能 ⟶ 传递

定义函数指针



函数指针

```c
void b() {
    printf("I am func b\n");
}//返回值是void 形参是空的
int main() {
    void(*p)();
    p = b;//函数名单独使用（后面不加圆括号），它的含义就是函数的入口地址
    p();
}
```



函数指针的传递

```
void a(void(*p)(int)) {
    p(3);
}
int main() {
    void(*p)(int);//定义的时候要加形参
    p = b;//函数名单独使用（后面不加圆括号），它的含义就是函数的入口地址
    //p(2);//调用的时候，传递实参
    a(p);
}
```