Ω表示算法最幸运时候的复杂度

数组的另外一种表示方法

```c
int numbers[] = {4, 6, 8, 2, 7, 5, 0};
```

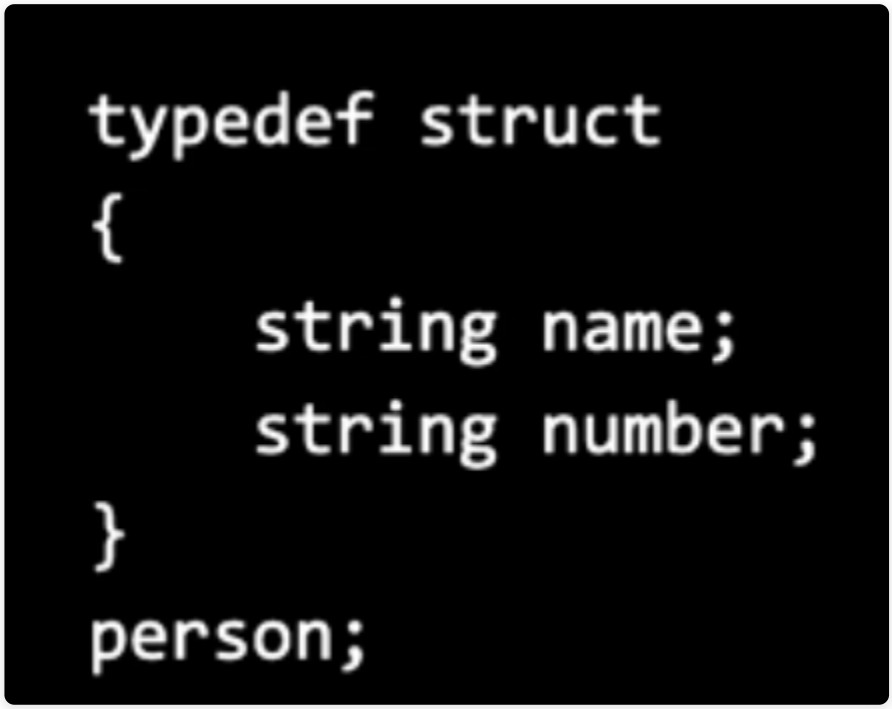
strcmp

即比较str数组中第i项是否为Ron

```c
strcmp(names[i], "Ron") == 0
```

*就，之前的部分在算法图解看过，就直接跳到structure了

数据结构



```
person people[2];

people[0].name = "Brian";
people[0].number = "+1-617-495-1000";

people[1].name = "David";
people[1].number = "+1-949-468-2750";
```

在后面会在头文件里定义

排序

是二分查找的基础

选择排序

循环。第一个循环找到最小值，放第二位。第二个循环找次小，放第二位，后面同理

o（n2）

冒泡排序

依次比较两个相邻数字的大小，把大的换到右边*有点类似滑窗

第一个循环保证最大的在最后，第二个循环保证次大在次后...

似乎也是o（n2）

但对于最优情况，Ω选择排序是n2，冒泡排序是n（查询一遍相邻的数字，发现都相等）

（递归） 归（合并）并排序

将问题划分为两半，左右分别排序，不断递归

然后再比较左右两侧最小的数字，放到大块儿的第一个，以此类推

O（nlogn），logn是是递归，n是合并，Ω也是

但空间复杂度高

*theta计数法：上下限一样的复杂度

算法

排序

搜索