

4、数据结构和算法——算法

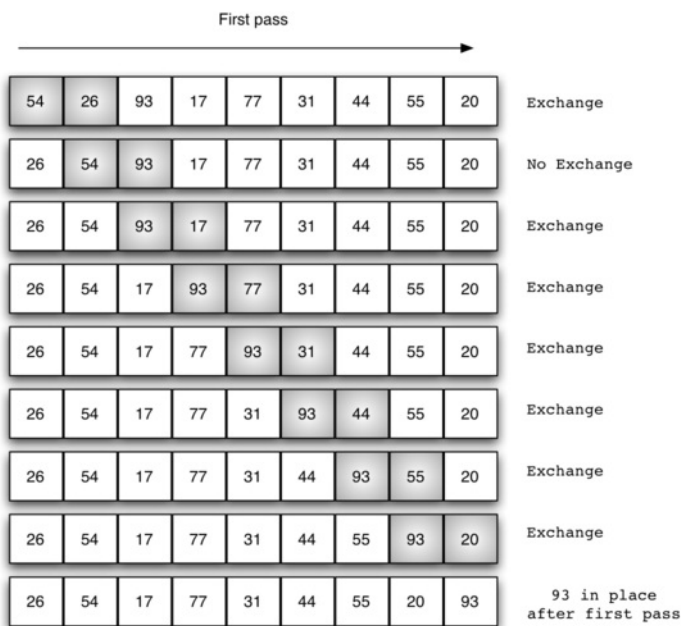
冒泡排序

冒泡排序（英语：Bubble Sort）是一种简单的排序算法。它重复地遍历要排序的数列，一次比较两个元素，如果他们的顺序错误就把他们交换过来。遍历数列的工作是重复地进行直到没有再需要交换，也就是说该数列已经排序完成。这个算法的名字由来是因为越小的元素会经由交换慢慢“浮”到数列的顶端。

冒泡排序算法的运作如下：

- 比较相邻的元素。如果第一个比第二个大（升序），就交换他们两个。
- 对每一对相邻元素作同样的工作，从开始第一对到结尾的最后一对。这步做完后，最后的元素会是最大的数。
- 针对所有的元素重复以上的步骤，除了最后一个。
- 持续每次对越来越少的元素重复上面的步骤，直到没有任何一对数字需要比较。

冒泡排序的分析



初始状态:

3	6	4	2	11	10	5
---	---	---	---	----	----	---

第1趟排序:

3	4	2	6	10	5	11
---	---	---	---	----	---	----

 (比较6次, 11沉到未排序序列尾部)

第2趟排序:

3	2	4	6	5	10	11
---	---	---	---	---	----	----

 (比较5次, 10沉到未排序序列尾部)

第3趟排序:

2	3	4	5	6	10	11
---	---	---	---	---	----	----

 (比较4次, 6沉到未排序序列尾部)

第4趟排序:

2	3	4	5	6	10	11
---	---	---	---	---	----	----

 (比较3次, 5沉到未排序序列尾部)

第5趟排序:

2	3	4	5	6	10	11
---	---	---	---	---	----	----

 (比较2次, 4沉到未排序序列尾部)

第6趟排序:

2	3	4	5	6	10	11
---	---	---	---	---	----	----

 (比较1次, 3沉到未排序序列尾部)

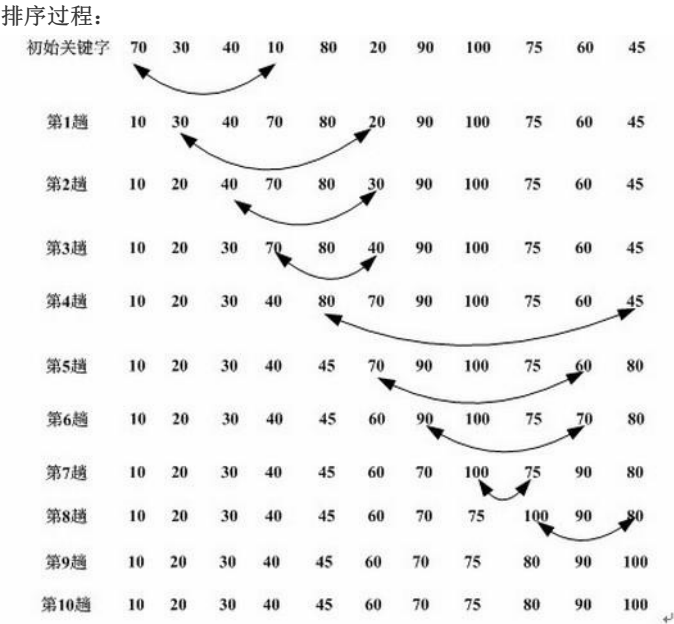
选择排序

选择排序（Selection sort）是一种简单直观的排序算法。它的工作原理如下。首先在未排序序列中找到最小（大）元素，存放到排序序列的起始位置，然后，再从剩余未排序元素中继续寻找最小（大）元素，然后放到已排序序列的末尾。以此类推，直到所有元

素均排序完毕。

选择排序的主要优点与数据移动有关。如果某个元素位于正确的最终位置上，则它不会被移动。选择排序每次交换一对元素，它们当中至少有一个将被移到其最终位置上，因此对n个元素的表进行排序总共进行至多n-1次交换。在所有的完全依靠交换去移动元素的排序方法中，选择排序属于非常好的一种。

选择排序分析

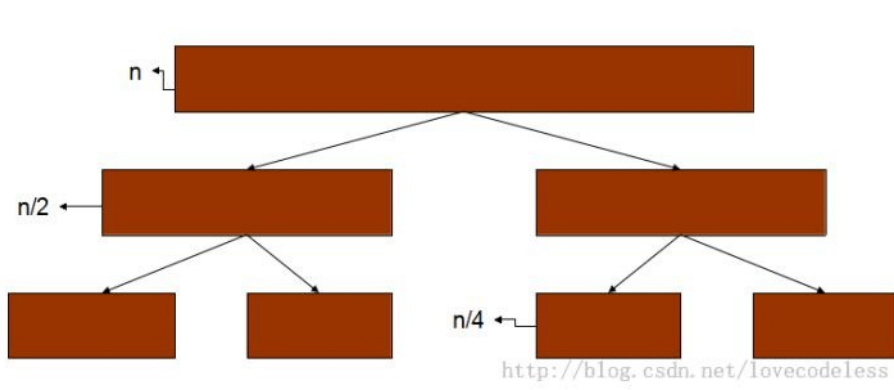


快速排序

快速排序（Quicksort）是对冒泡排序的一种改进。

快速排序由C. A. R. Hoare在1962年提出。它的基本思想是：通过一趟排序将要排序的数据分割成独立的两部分，其中一部分的所有数据都比另外一部分的所有数据都要小，然后再按此方法对这两部分数据分别进行快速排序，整个排序过程可以递归进行，以此达到整个数据变成有序序列。





算法的稳定性：

$[5(a), 3, 4, 5(b), 2, 10] \rightarrow [2, 3, 4, 5(b), 5(a), 10]$ ：保持原始列表中的同元数的顺序则为稳定，否则为不稳定。

冒泡排序是稳定的算法，选择排序是不稳定的算法

搜索

搜索是在一个项目集中找到一个特定项目的算法过程。搜索通常的答案是真的或假的，因为该项目是否存在。搜索的几种常见方法：顺序查找、二分法查找、二叉树查找、哈希查找

二分法查找

当数据量很大适宜采用该方法。采用二分法查找时，数据需是排好序的。

假如有一组数为3，12，24，36，55，68，75，88要查给定的值24.可设三个变量front，mid，end分别指向数据的上界，中间和下界， $mid = (front + end) / 2$ 。

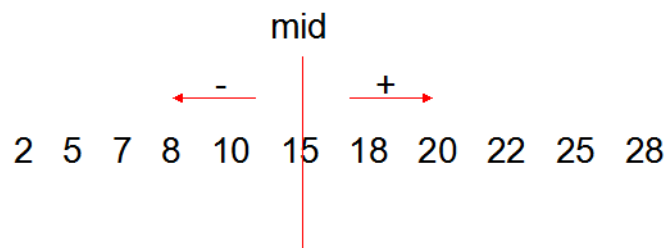
1.开始令front=0（指向3），end=7（指向88），则mid=3（指向36）。因为mid>x，故应在前半段中查找。

2.令新的end=mid-1=2，而front=0不变，则新的mid=1。此时x>mid，故确定应在后半段中查找。

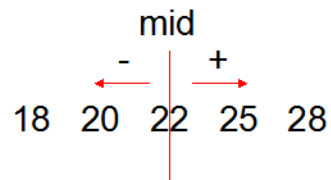
3.令新的front=mid+1=2，而end=2不变，则新的mid=2，此时a[mid]=x，查找成功。

如果要查找的数不是数列中的数，例如x=25，当第三次判断时， $x > a[mid]$ ，按以上规律，令front=mid+1，即front=3，出现front>end的情况，表示查找不成功。

step 1:



step 2:



step 3:

