

冒泡排序

1 冒泡排序

- 依次比较相邻的两个记录的关键字，若两个记录是反序的，即前一个记录的关键字大于后一个记录的关键字，则进行交换，否则不动，直到没有反序的记录为止。

1 冒泡排序

- (1) 首先将 $R[1]$ 和 $R[2]$ 的关键字进行比较，若为反序，则交换这两个记录，否则不动。再比较 $R[2]$ 和 $R[3]$ 的关键字，若为反序，则交换这两个记录，否则不动。依此类推，直到比较 $R[n-1]$ 和 $R[n]$ 的关键字并处理后为止。这是第一趟冒泡排序， $R[n]$ 一定是关键字最大的记录。

1 冒泡排序

- (2) 然后进行第2趟冒泡排序，对前 $n-1$ 个记录进行同样的处理。依此类推，直到进行完第 $n-1$ 趟冒泡排序为止。


1 冒泡排序

- 设记录序列为60,70,50,30,80,20,10,40。


1 冒泡排序

- 60 70 50 30 80 20 10 40
- 
- The diagram illustrates the first pass of bubble sort on the array [60, 70, 50, 30, 80, 20, 10, 40]. Blue curved arrows indicate the following comparisons and swaps: (70, 50), (50, 30), (80, 20), (20, 10), and (10, 40). The element 60 remains in its original position as it is already in the sorted order relative to the elements it is compared with.

1 冒泡排序

- 60 70 50 30 80 20 10 40
 - 60 50 30 70 20 10 40 80
- 

1 冒泡排序

- 60 70 50 30 80 20 10 40
 - 60 50 30 70 20 10 40 80
 - 50 30 60 20 10 40 70 80
- 

1 冒泡排序

- 60 70 50 30 80 20 10 40
 - 60 50 30 70 20 10 40 80
 - 50 30 60 20 10 40 70 80
 - 30 50 20 10 40 60 70 80
- 

1 冒泡排序

- 60 70 50 30 80 20 10 40
 - 60 50 30 70 20 10 40 80
 - 50 30 60 20 10 40 70 80
 - 30 50 20 10 40 60 70 80
 - 30 20 10 40 50 60 70 80
- 

1 冒泡排序

- 60 70 50 30 80 20 10 40
 - 60 50 30 70 20 10 40 80
 - 50 30 60 20 10 40 70 80
 - 30 50 20 10 40 60 70 80
 - 30 20 10 40 50 60 70 80
 - 20 10 30 40 50 60 70 80
- 

1 冒泡排序

- 60 70 50 30 80 20 10 40
- 60 50 30 70 20 10 40 80
- 50 30 60 20 10 40 70 80
- 30 50 20 10 40 60 70 80
- 30 20 10 40 50 60 70 80
- 20 10 30 40 50 60 70 80
- 10 20 30 40 50 60 70 80

1 冒泡排序

- 60 70 50 30 80 20 10 40
- 60 50 30 70 20 10 40 80
- 50 30 60 20 10 40 70 80
- 30 50 20 10 40 60 70 80
- 30 20 10 40 50 60 70 80
- 20 10 30 40 50 60 70 80
- 10 20 30 40 50 60 70 80
- 10 20 30 40 50 60 70 80

2 改进版冒泡排序

- 在进行冒泡排序过程中，如果某一趟排序，发现相邻记录全部都是正序的话，那么说明所有记录已经排好序，可以提前结束。

思考

- 随机产生10000个整数，分别用冒泡排序和改进版冒泡排序对它们进行排序，比较运行的时间。