冒泡排序：

老版冒泡排序为用循环在一组数据中每次找出最大值（或最小值），然后再次循环找出其次的最大值（或最小值），最终达成排序的效果

如：

Int a[10],i,j,num; //定义数组与变量

for(i=0;i<10;i++) //每次循环可将最大数以下依次排好一个

For(j=0;j<9;j++) //此循环负责找出除排好以外的最大数

If(a[j]>a[j+1]) //交换数值

{num=a[j];a[j]=a[j+1];a[j+1]=num;

}

改进几次后的冒泡，可以在一次循环中找出最大值与最小值，耗时变短

int a[10],low=0，high=9,i,num; //定义数组与变量

while (low<high)

{

        for (i=low; i<high; ++i) //从左到右找出最大值

            if (arr[i]> arr[i+1])

{

                num=arr[i]; arr[i]=arr[i+1];arr[i+1]=num;

   }

        --high;                 //最大值已经找到，high前移一位

        for (i=high; i>low; --i) //从右到左排序找出最小值           if (arr[i]<arr[i-1])

{

                num=arr[i]; arr[i]=arr[i-1];arr[i-1]=num;

            }

        ++low;                  //最小值已经找到，low向后移动

}

循环结束后便排好序，且耗时比老版冒泡短

插入排序：

插入排序类似扑克牌整理，将没有排序的数据去和已经排好的数据比较然后插入其中，最后达到排序的效果

int a[10],i,j,num; //定义数组与变量

for (i=1; i< 10;i++) //要进行比较的次数

{

            num=a[i]; //要插入的数

            j=i-1;

            while(j>=0&&a[j]>num)

//将已经排好的数据右移为要插入的地方腾出位置

{

                a[j+1]=a[j];

                j--;

            }

            a[j+1]=num; //插入数据

        }

插入排序改进后为二分法，即把未排序的数与已排序的数中间值做比较找出其应该在中间值前面还是后面，已排序的部分则分为两部分，然后再与其对应部分中间值做比较，以此类推，可以减少耗时

int a[10],i,j,num,left,right,middle; //定义数组与变量

for (i=1;i<10;i++)

{

            num=a[i],left =0,right=i-1;

            while (left<=right) //不断寻找中间值最后找到相应位置

{

                middle=(left+right) / 2;

                if (num<a[middle])

                    right=middle-1;

                else

                    left=middle+1;

    }

            for (j=i-1; j>=left; j--)

                a[j+1]=a[j]; //将排好数据右移腾出位置

            a[left]=num; //将数据放入腾出的位置

        }