数据结构与算法3-插入排序

笔记本:	我的笔记		
创建时间:	2020/9/18 21:58	**更新时间:**	2020/10/3 10:05
作者:	liuhouer		
标签:	算法		

基础排序算法：插入排序

arr[0, i)已排好序; arr[i.n) 未排序
把ar[i]放到合适的位置

时间复杂度：O(n^2)


```
public class InsertionSort {

    private InsertionSort(){}

    public static <E extends Comparable<E>> void sort(E[] arr){

        for(int i = 0; i < arr.length; i ++){
```

```
            // 将 arr[i] 插入到合适的位置
//            for(int j = i; j - 1 >= 0; j --){
//                if(arr[j].compareTo(arr[j - 1]) < 0)
//                    swap(arr, j - 1, j);
//                else break;
//            }

            for(int j = i; j - 1 >= 0 && arr[j].compareTo(arr[j - 1]) < 0; j --)
                swap(arr, j - 1, j);
        }
    }

    private static <E> void swap(E[] arr, int i, int j){

        E t = arr[i];
        arr[i] = arr[j];
        arr[j] = t;
    }

    private static <E extends Comparable<E>> boolean isSorted(E[] arr){

        for(int i = 1; i < arr.length; i ++)
            if(arr[i - 1].compareTo(arr[i]) > 0)
                return false;
        return true;
    }

    public static void main(String[] args){

        int[] dataSize = {10000, 100000};
        for(int n: dataSize){
            Integer[] arr = ArrayGenerator.generateRandomArray(n, n);
            SortingHelper.sortTest("InsertionSort", arr);
        }
    }
}
```

优化插入排序，每次和前面的比较值，不是比较一次更换一次位置，而是通过一个序号往前找，找到以后，后面的从后往前位移一位，只替换值即可，完成后把序号的位置的值更换为待替换的值即可

```
public class SelectionSort {

    private SelectionSort(){}

    public static <E extends Comparable> void sort(E[] arr){

        for(int i = 0; i < arr.length; i ++){

            // 选择 arr[i...n) 中的最小值
            int minIndex = i;
            for(int j = i; j < arr.length; j ++){
                if(arr[j].compareTo(arr[minIndex]) < 0)
                    minIndex = j;
            }
```

```
            swap(arr, i, minIndex);
        }
    }


    private static <E> void swap(E[] arr, int i, int j){


        E t = arr[i];
        arr[i] = arr[j];
        arr[j] = t;
    }


    public static void main(String[] args){


        int[] dataSize = {10000, 100000};
        for(int n: dataSize){
            Integer[] arr = ArrayGenerator.generateRandomArray(n, n);
            SortingHelper.sortTest("SelectionSort", arr);
        }
    }
}
```

选择排序法
插入排序法
循环不变量
均是 O(n^2) 算法
对于完全有序的数组，插入排序成为 O(n) 的算法【绝大部分元素不用位移，仅仅查找一位，所以是O(n)】