山东大学 计算机科学与技术 学院

数据结构与算法 课程实验报告

学号：202200130048	姓名： 陈静雯	班级： 6
实验题目：排序算法		
实验学时：2	实验日期：9.28	

实验目的：

1. 名次排序、及时终止的选择排序、及时终止的冒泡排序、插入排序

软件开发工具：

Vscode

2. 实验内容

   1. 不得使用与实验相关的 STL
   2. 需使用类模版(template<class T>)
   3. 需定义排序类，封装各排序方法
   4. 排序数据需使用动态数组存储
   5. 排序类需提供以下操作：名次排序、及时终止的选择排序、及时终止的冒泡排序、插入排序

3. 数据结构与算法描述 （整体思路描述，所需要的数据结构与算法）

   （1）名次排序：将数组中的元素两两比较，较大的元素名次加一，直到所有都比较完，得到元素的名次数组，利用名次数组对原数组进行原地重排，即用循环不断对该位置的数进行交换，直到名次与位置对应

   （2）及时终止的选择排序：遍历数组找到最大值将其放到数组最后，如果遍历是 temp 的值一直交换，说明数组已有序，可退出循环

   （3）及时终止的冒泡排序：从前往后遍历数组，如果前一个比后一个大就与之交换，若遍历时没有交换，说明数组有序。

   （4）插入排序：被插入数与前面的有序数组一个个比较，比元素小，被比较数就往后移，直到比被比较的元素大，插入到合适位置。

4. 测试结果（测试输入，测试输出）

5. 分析与探讨（结果分析，若存在问题，探讨解决问题的途径）
相比之下，冒泡排序效率较高，名次和插入都不能即使终止，时间复杂度都为 0(n 方)。

6. 附录：实现源代码（本实验的全部源程序代码，程序风格清晰易理解，有充分的注释）

```
#include <iostream>
using namespace std;

template<class T>
void myswap(T& a,T& b){ //交换函数
    T temp=a;
    a=b;
    b=temp;
}

template<typename T>
class mySort{
public:
    mySort(int n,T a[]){  //构造函数把数组和 n 赋给指针 p 和 num
        p=new T [n];
        for(int i=0;i<n;i++) p[i]=a[i];
        num=n;
    }
    void sort_rank();
    void sort_select();
    void sort_bubble();
    void sort_insert();
    void show();
    ~mySort(){delete [] p;}
private:
    T* p;  //待排序数组
    int num; //数组元素个数
};

template<typename T>
void mySort<T>::sort_rank(){   //名次排序
    int r[1005]={0};
    for(int i=0;i<num;i++){
        for(int j=i+1;j<num;j++){
            if(p[i]<=p[j]) r[j]++; //计算名词，哪个大哪个名词+1
            else r[i]++;
        }
    }
    for(int i=0;i<num;i++){
        while(r[i]!=i){
            int t=r[i];
```

```
            myswap(p[i],p[t]);  //原地重排，不断循环直到该位置是该名词的数
            myswap(r[i],r[t]);
        }
    }
}

template<typename T>
void mySort<T>::sort_select(){  //选择排序
    bool check=false;
    int last=num;
    while(!check&&last>1){
        int temp=0;
        check=true;
        for(int i=1;i<last;i++){
            if(p[temp]<=p[i]) temp=i; //寻找最大值，若每次循环都改变最大值下标说明数组从小到大有序
            else check=false;           //数组无序
        }
        myswap(p[temp],p[--last]);//把最大值放最后
    }
}

template<typename T>
void mySort<T>::sort_bubble(){ //冒泡排序
    bool check=false;
    int last=num;
    while(!check&&last>1){
        check=true;
        for(int i=0;i<last-1;i++){
            if(p[i]>p[i+1]){
                myswap(p[i],p[i+1]);//有交换标记为 false，无交换说明数组有序
                check=false;
            }
        }
        last--;
    }
}

template<typename T>
void mySort<T>::sort_insert(){ //插入排序
    for(int i=1;i<num;i++){
        int temp=p[i];
        int j=i-1;
        for( ;j>=0;j--){
            if(temp<p[j]){ //待插入数小，被比较数往后移
                p[j+1]=p[j];
```

```
            }
            else{
                break;
            }
        }
        p[j+1]=temp; //将待插入数放入空位
    }
}

template<typename T>
void mySort<T>::show(){  //输出有序数组
    for(int i=0;i<num;i++){
        cout<<p[i]<<" ";
    }
    cout<<endl;
}

int main(){
    int r[1005]={0};
    int a[1005]={0};
    int n;
    cin>>n;
    for(int i=0;i<n;i++) cin>>a[i];
    mySort<int> A(n,a);
    A.sort_select();
    A.show();
}
```