

数据结构与算法 课程实验报告

学号：202000130143	姓名： 郑凯饶	班级： 计科 20.1
实验题目：排序算法		
实验学时：2	实验日期： 1006	
实验目的： 掌握各种简单排序算法		
软件开发环境： Windows 10 家庭中文版 64 位 (10.0, 版本 18363) Dev-C++ IDE		
<p>1. 实验内容</p> <p>设计排序类，使用模板类设计，定义排序类，封装各种排序方法，并且排序数据使用动态数组储存。</p> <p>2. 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）</p> <p>对一组数据创建一个排序类的实例，并且可以调用各种方法进行排序。</p> <p>析构时，析构动态创建的数组空间，防止内存泄露。</p> <p>3. 测试结果（测试输入，测试输出）</p> <p>在 OJ 平台上成功提交。</p> <p>4. 分析与探讨（结果分析，若存在问题，探讨解决问题的途径）</p> <p>一开始，对如何传递数组的写法比较迷糊~</p> <p>搜索知道有以下两种写法：</p> <pre>Sort(T *P, int n) { // T p[] this->p = P; this->n = n; }</pre> <p>5. 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）</p> <pre>1. #include<bits/stdc++.h> 2. using namespace std; 3. 4. // 数组对象， 数组长度 5. // 0 - n-1 6. template<class T> 7. class Sort { 8. private: 9. int n; 10. T *p;</pre>		

```

11. public:
12. Sort(T *P, int n) { // T p[]
13.     this->p = P;
14.     this->n = n;
15. }
16. ~Sort() {
17.     delete [] p;
18. }
19. void Rank_sort();
20. void Selection_sort();
21. void Bubble_sort();
22. void Insection_sort();
23. void output() {
24.     for (int i=0; i<n; i++) cout << p[i] << ' ';
25.     cout << '\n';
26. }
27.};
28.
29.
30.template<class T>
31.void Sort<T>::Rank_sort() {
32.    // 名次计算
33.    int *r = new int [n];
34.    for (int i=0; i<n; i++) r[i] = 0;
35.
36.    for (int i=1; i<n; i++) {
37.        for (int j=0; j<i; j++) {
38.            if(p[j] <= p[i]) r[i] ++;
39.            else r[j] ++;
40.        }
41.    }
42.
43.    T *u = new T [n];
44.    for (int i=0; i<n; i++) u[r[i]] = p[i];
45.    for (int i=0; i<n; i++) p[i] = u[i];
46.    delete [] u;
47.    delete [] r;
48.}
49.
50.template<class T>
51.void Sort<T>::Selection_sort() {
52.    bool sorted = false;
53.    for (int i = n; !sorted && (i > 1); i--) {
54.        int IndexOfMax = 0;
55.        sorted = true;
56.        for (int j = 1; j < i; j++) {

```

```

57.     if(p[IndexOfMax] <= p[j]) IndexOfMax = j;
58.     else sorted = false;
59.     swap(p[IndexOfMax], p[i - 1]);
60. }
61. }
62.}
63.
64.template<class T>
65.void Sort<T>::Bubble_sort() {
66.    for (int i = n, swaped; i > 1 && swaped; i--) {
67.        swaped = false;
68.        for (int j = 0; j < i - 1; j++) {
69.            if(p[j] > p[j+1]){
70.                swap(p[j], p[j+1]);
71.                swaped = true;
72.            }
73.        }
74.    }
75.}
76.
77.template<class T>
78.void Sort<T>::Insection_sort() {
79.    for (int i = 1; i < n; i++) {
80.        T t = p[i];
81.        int j;
82.        for (j = i - 1; j >= 0 && t < p[j]; j --)
83.            p[j+1] = p[j];
84.        p[j+1] = t;
85.    }
86.}
87.
88.int main(){
89.    int n; cin >> n;
90.    int *a = new int [n];
91.    for (int i=0; i<n; i++) cin >> a[i];
92.
93.    Sort<int> A(a, n); // 传递 a[] 首地址
94.    // A.Rank_sort();
95.    // A.Selection_sort();
96.    // A.Bubble_sort();
97.    A.Insection_sort();
98.    A.output();
99.
100.    return 0;
101.}

```

