**附件：**

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：202000130138 | 姓名：宋璎航 | | 班级： 20.3 |
| 实验题目：排序算法 | | | |
| 实验学时：2 | | 实验日期： 2021/10/7 | |
| 实验目的：  掌握各种简单排序算法 | | | |
| 软件开发环境：  VSC | | | |
| 1. 实验内容   **题目描述**：  用任意一种排序方式给出n个整数按升序排序后的结果，满足以下要求：  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。  **输入输出格式**：  **输入**：输入的第一行是一个整数n(1<=n<=1000)，表示需排序的数的个数。接下来一行是n个整数，数的范围是0到1000，每两个相邻数据间用一个空格分隔。  **输出**：一行排好序的序列。   1. 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）   建立排序类Sort，分别构建选择排序、插入排序、冒泡排序、名次排序四种排序方法函数，其中选择排序与冒泡排序及时终止   1. 测试结果（测试输入，测试输出）   输入：  5  5 4 3 2 1  输出：  1 2 3 4 5   1. 分析与探讨（结果分析，若存在问题，探讨解决问题的途径）   结果均正确   1. 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* <bits/stdc++.h>  using namespace std;  template<class T>  class Sort  {      T \*a;      int n;  public:      Sort() {}      Sort(int n)  {  this->n = n;          a = new T[n];  *for* (int i = 0; i < n; ++i)              cin >> a[i];      }        void output()      {  *for* (int i = 0; i < n; ++i)              cout << a[i] << ' ';      }      void rank\_sort()*//名次排序*      {          int \*r;          r = new int[n];  *for* (int i = 0; i < n; ++i)              r[i] = 0;  *for* (int i = 1; i < n; ++i)  *for* (int j = 0; j < i; ++j)  *if* (a[j] <= a[i])                      r[i]++;  *else*                      r[j]++;          int \*temp;          temp = new T[n];  *for* (int i = 0; i < n; ++i)              temp[r[i]] = a[i];  *for* (int i = 0; i < n; ++i)              a[i] = temp[i];          delete[] temp;      }      void select\_sort()*//选择排序*      {          bool sorted = false;          int p = n;  *while* (!sorted)          {              int maxpos = 0;              sorted = true;  *for* (int i = 1; i < p; ++i)              {  *if* (a[i] >= a[maxpos])                      maxpos = i;  *else*                      sorted = false;              }              swap(a[maxpos], a[--p]);          }      }      void bubble\_sort()*//冒泡排序*      {  *for* (int i = 1; i < n; ++i)          {              bool swapped = false;  *for* (int j = n - 1; j >= i; --j)  *if* (a[j] < a[j - 1])                  {                      swap(a[j], a[j - 1]);                      swapped = true;                  }  *if* (!swapped)  *return*;          }      }      void insert\_sort()*//插入排序*      {  *for* (int i = 0; i < n; ++i)          {              T temp = a[i];              int j = 0;  *for* (j = i - 1; j >= 0 and temp < a[j]; --j)                  a[j + 1] = a[j];              a[j + 1] = temp;          }      }      ~Sort() { delete[] a; }  };  int main()  {      int n=0;      cin >> n;      Sort<int> s(n);      s.rank\_sort();      s.output();  } | | | |