**Sort Comprehensive Experiment Report**

Class:计科201 Student ID 1:20401010101 Name 1:沈梦越 Experiment Date:2021/12/8

Student ID 2:20401010114 Name 2:余亚宁

Student ID 3:20401010106 Name 3:杨婧菡

**One. Experimental purpose**

1) Familiar with the basic operations of the sort.2) Master the operation of various internal sorting.3) Deepen the understanding of the sort, and to develop the programming ability of solving practical problems gradually.

**Two. Experimental environment**

Computers equipped with Visual C6.0/CFree.The experiment lasted for 4 hours.

**Three. Experimental content**

A series of strings are stored in a two-dimensional array. Try to sort them with some sorting algorithms (at least two algorithms, such as insert sorting, bubble sorting, quick sorting, and heap sorting). You should sort them to dictionary order finally.For example: two-dimensional array is :char s[][20]={“while”，”if”，“else”，”do”，“for”，”switch”，“case”};

**Four. Important data structures**

void InsertSort(string a[], int n)

{

for (int j = 1; j < n; j++)

{

string key = a[j];

int i = j - 1;

while (i >= 0 && key < a[i])

{

a[i + 1] = a[i];

i--;

}

a[i + 1] = key;

}

}

void BubbleSort(int arr[], int n)

{

for (int i = 0; i < n - 1; i++)

{

for (int j = 0; j < n - i - 1; j++)

{

if (arr[j] > arr[j + 1])

std::swap(arr[j], arr[j + 1]);

}

}

}

**Five. Realization idea analysis**

It is easy to understand that when we play poker, every time we touch the cards, we will insert them into the appropriate position according to the number size or color until we touch the last card, and the cards in our hands have been arranged according to the size order. The whole process is an insertion sort

Bubble sorting is a simple sorting algorithm. It repeatedly visits the element column to be sorted, compares two adjacent elements in turn, and exchanges them if they are in the wrong order. The work of visiting elements is repeated until no adjacent elements need to be exchanged, that is, the elements have been sorted.

**Six. Program debugging problem analysis**

Problems encountered in debugging and Solutions

**Seven. Experimental summary**

#include<iostream>

#include<fstream>

#include<string>

using namespace std;

void InsertSort(string a[], int n)

{

for (int j = 1; j < n; j++)

{

string key = a[j];

int i = j - 1;

while (i >= 0 && key < a[i])

{

a[i + 1] = a[i];

i--;

}

a[i + 1] = key;

}

}

void BubbleSort(int arr[], int n)

{

for (int i = 0; i < n - 1; i++)

{

for (int j = 0; j < n - i - 1; j++)

{

if (arr[j] > arr[j + 1])

std::swap(arr[j], arr[j + 1]);

}

}

}

void main()

{

int i = 0;

string d[10];

ifstream infile;

infile.open("text.txt");

while (!infile.eof())

{

infile >> d[i];

if (infile.eof())

break;

i++;

}

infile.close();

InsertSort(d, i);

cout << "排序后结果：";

for (int j = 0; j < i; j++)

{

cout << d[j] << " ";

}

}

**Eight. Crew Division**

|  |  |  |
| --- | --- | --- |
| **Group division** | | |
| **Member name** | **Work done** | **Completion situation** |
| **沈梦越** | **报告** | **100%** |
| **余亚宁** | **代码** | **100%** |
| **杨婧菡** | **代码** | **60%** |