LAB 02. THUẬT GIẢI SẮP XẾP TRONG

A. Mục tiêu
B. Yêu cầu
C. Ôn tập
D. Luyện tập
E. Bài tập

A. Mục tiêu

Cài đặt các thuật giải sắp xếp trong

B. Yêu cầu thực hành

- Tạo thư mục, đặt tên là **MaSV_Lab02** để lưu bài làm. Trong đó, MaSV là mã số của sinh viên.
- Sinh viên phải nạp phần luyện tập cho GV Tại phòng Lab (vào cuối buổi thực tập theo thông báo của giáo viên).
- Sinh viên nạp bài tập 1,3 trong phần bài tập cho giáo viên tại phòng lab (vào cuối giờ thực theo thông báo của giáo viên).

C. Ôn tập

a. Phát biểu bài toán :
Cho dãy a[0,…,n-1] gồm n số, sắp dãy này tăng dần.

- Input : $a_0, a_2, \ldots a_{n-1}$
  int a[n];
- Output :
  - a tảng dần

Ghi chú : Tổng quát, sắp tăng một mảng cấu trúc theo một trường dữ liệu nào đó (có thứ tự).

b. Các thuật giải sắp xếp trong:
1. Chọn trực tiếp (Selection Straight sort)
2. Chèn trực tiếp (Insertion Straight sort)
3. Đổi chỗ trực tiếp (Interchange Straight sort)
4. Nổi bọt (Bubble sort)
5. Chèn nhị phân (Binary Insertion Sort)
6. Quick sort
7. Heap Sort
8. Merge Sort
9. Radix Sort

c. Ý tưởng các thuật giải :

1. Chọn trực tiếp (Selection Straight sort)

Thuật giải tiến hành trong n-1 bước, Với mọi i = 0,…, n-2 , tìm min trong đoạn $[a_i, a_{n-1}]$ rồi đưa Min về đầu dãy $a_i,a_{i+1},\ldots,a_{n-1}$

2. Chèn trực tiếp (Insertion Straight sort):

Thuật giải tiến hành trong n-1 bước, Với mọi i = 1,…, n-1 , tìm vị trí thích hợp của dãy con tăng dần $a_0,a_{i+1},\ldots,a_{i-1}$ để chèn a[i] vào tạo ra dãy con i+1 phần tử $a_0,a_{i+1},\ldots,a_i$ tăng dần.
Tìm vị trí thích hợp này bằng thuật giải tím kiếm tuyến tính từ vị trí i-1 về 0.

3. Đổi chỗ trực tiếp (Interchange Straight sort):

Thuật giải tiến hành trong n-1 bước, Với mọi i = 0,…, n-2, tìm các phần tủ sau a_i và tạo với a_i thành một cặp nghịch thế, rồi triệt tiêu các cặp nghịch thế này.

4. Nổi bọt (Bubble sort):

Thuật giải tiến hành trong n-1 bước, Với mọi bước i = 0,…, n-2, xuất phát từ cuối dãy đổi chỗ các cặp phần tử kế cận để đưa phần tử nhỏ nhất về đầu dãy .

5. Chèn nhị phân (Binary Insertion Sort):

Thuật giải tiến hành trong n-1 bước, Với mọi i = 1,…, n-1 , tìm vị trí thích hợp của dãy con tăng dần $a_0,a_{i+1},\ldots,a_{i-1}$ để chèn a[i] vào tạo ra dãy con i+1 phần tử $a_0,a_{i+1},\ldots,a_i$ tăng dần.
Tìm vị trí thích hợp này bằng thuật giải tím kiếm nhị phân

6. Thuật giải Quick sort:

Cải tiến thuật giải đổi chỗ trực tiếp bằng cách đổi chỗ các cặp không đúng thứ tự có thể xa nhau.

Thuật toán thực hiện việc lặp phân đoạn mảng đã cho thành 2 phần, nửa mảng bên trái <= x, nửa mảng bên phải >= x, cho đến khi các mảng con chỉ có 1 phần tử.

7. Thuật giải Heap Sort:

Cải tiến thuật giải chọn trực tiếp bằng cách tại mỗi bước, tận dụng thông tin các phép toán so sánh ở bước trước.
Từ heap ban đầu phát triển thành heap đầy đủ ban đầu, hoán vị 2 phần tử đầu và cuối để đưa giá trị lớp nhất của dãy về cuối dãy. Loại phần tử cuối (giá trị lớn nhất), bổ sung a_0 vào heap gồm các phần từ sau đó để tạo thành một heap, rồi lặp lại các đua giá trị lớn nhất về cuối.

Lặp quá trình trên khi dã chỉ còn 1 phần tử.
8. Thuật giải Merge Sort:

Thuật giải tiến hành nhiều bước lặp, mỗi bước gồm 2 giai đoạn :
- Tách luân phiên p phần tử của dãy đã cho vào 2 dãy trung gian.
- Trộn từng p phần tử của 2 dãy trung gian để tạo ra 2p phần tử tang dần rồi lưu vào dãy ban đầu.
- p khởi tạo ban đầu bằng 1. Tại mỗi bước p khởi tạo lại bằng 2p.

Thuật giải kết thúc khi p > n.

9. Thuật giải Radix sort:

Mô phỏng cách phân phối thư của bưu điện.

D. LUYỆN TẬP

Bài 1:

Viết chương trình tùy chọn sắp tăng dãy n số nguyên với các thuật giải:

1. Chọn trực tiếp (Selection Straight sort)
2. Chèn trực tiếp (Insertion Straight sort)
3. Đổi chỗ trực tiếp (Interchange Straight sort)
4. Nổi bọt (Bubble sort)
5. Chèn nhị phân (Binary Insertion Sort)
6. Radix

- Tập tin dữ liệu "***text1.txt***":

8 //Kich thuoc mang
12 2 8 5 1 6 4 15
- Tập tin dữ liệu "***text1.txt***":

12 //Kich thuoc mang
7013 8425 1239 428 1424 7009 4518 3252 9170 999 1725 701

Thực hiện:

Bước 1. Tạo dự án Win32 Console Application mới. Đặt tên là **Lab04_Bai01**

Bước 2. Tạo các tập tin rỗng sau trong project **Lab04_Bai01:**

- Tạo các tập tin thư viện sau trong **Header Files : menu.h, thuvien.h**
- Tạo tập tin chương trình trong **Source File : Program.cpp**

Bước 3. Soạn thảo từng phần cho các tập tin trên. Sau mỗi phần ta chạy chương trình kiểm tra kết quả công việc đã làm trong phần đó.

Nội dung tối thiểu được dùng để phát triển dự án mà chương trình có thể chạy được:

- Trong tập tin **Program.cpp** nhập nội dung sau

```
#include <iostream>
using namespace std;
#include "ThuVien.h"
#include "Menu.h"
void ChayChuongTrinh();

//=============================================
void ChayChuongTrinh()
{
	cout << "\nChuong trinh dang cap nhat.\n";
	system("PAUSE");
}

int main()
{
	ChayChuongTrinh();
	return 1;

}
```

Chạy chương trình để kiểm tra kết quả và sửa lỗi nếu có.

Bước 4: Hoàn thiện tiếp các tập tin trong project :

- Soạn thảo nội dung sau trong **menu.h** (tạo hệ thống menu):

```
//soan thao he thong menu
//=======================================================
//Hàm xuất menu, hàm này cấu trúc không đổi trong các project, chỉ khác tên chức năng
void XuatMenu()
{
	cout << "\n================ He thong chuc nang ===============";
	cout << "\n0. Thoat khoi chuong trinh";
	cout << "\n1. Chon Truc tiep - tai moi buoc dua GTNN ve dau mang";
	cout << "\n2. Chen Truc tiep - chen vao day con tang ben trai";
	cout << "\n3. Doi cho Truc tiep - tai moi buoc dua GTNN ve dau mang";
	cout << "\n4. Buble - tai moi buoc dua GTNN ve dau mang";
	cout << "\n5. Chen nhi phan";
	cout << "\n6. Radix Sort";
	cout << "\n9. Xem du lieu dang su dung";
	cout << "\n10. Chon du lieu khac";
}
```

Chạy chương trình để kiểm tra kết quả và sửa lỗi nếu có.

```
//Hàm này thường không đổi.
int ChonMenu(int soMenu)
```

```
{
	int stt;
	for (;;)
	{
		system("CLS"); //xoa man hinh
		XuatMenu();
		cout << "\nNhap mot so ( 0 <= so <= " << soMenu << " ) de chon menu, stt = ";
		cin >> stt;
		if (0 <= stt && stt <= soMenu)
			break;
	}
	return stt;
}

void XuLyMenu(int menu)
{	switch (menu)
	{
	case 0:
		system("CLS");
		cout << "\n0. Thoat khoi chuong trinh\n";
		break;
	case 1:
		system("CLS");
		cout << "\n1. Chon Truc tiep - tai moi buoc dua GTNN ve dau mang";
		break;
	case 2:
		system("CLS");
		cout << "\n2. Chen Truc tiep - chen vao day con tang ben trai";
		break;
	//…
	}
	system("PAUSE");
}
```

Chạy chương trình để kiểm tra kết quả và sửa lỗi nếu có.

- Trong tập tin Program.cpp cập nhật lại nội dung hàm **ChayChuongTrinh()** sao cho điều khiển được việc lặp thực hiện các chức năng với menu đã chọn (và dừng chương trình nếu chọn 0)

```
void ChayChuongTrinh()
{
	int soMenu = 10,
		menu;
	{
		system("CLS");
		menu = ChonMenu(soMenu);
		XuLyMenu(menu);
	} while (menu > 0);
}
```

Chạy chương trình để kiểm tra kết quả và sửa lỗi nếu có.

Bước 5 : Phát triển dần project. Soạn thảo tiếp các hàm chức năng trong **Thuvien.h**, soạn thảo các hàm xuất dữ liệu, nhập dữ liệu từ tập tin (chuẩn bị tập dữ liệu) sửa đổi lại hàm **XuLyMenu** trong **menu. h** cho phù hợp (có dữ liệu), sửa đổi lại hàm **ChayChuongTrinh** và bổ sung các khai báo thu viện trong **Program.cpp**.

- Trong Thuvien.h, ta định nghĩa hằng cho khai báo mảng, soạn thảo hàm dữ liệu từ tập tin văn bản vào

mảng 1 chiều các số nguyên (số nguyên hàng trên của tập tin là kích thước của mảng dữ liệu), xuất dữ liệu và một hàm hàm chức năng, chẳng hạn chọn trực tiếp

```
//Định nghĩa hằng
#define MAX 1000

//Khai báo nguyên mẫu các hàm

//Đọc dữ liệu trong tập tin filename chuyển qua mảng a
void File_Array(char *filename, int a[MAX], int &n)
{
	ifstream in(filename);
	if (!in)
	{
		cout << "\nLoi mo file !";
		exit(1);
	}
	in >> n;

	for (int i = 0; i < n; i++)
	{
		in >> a[i];

	}
	in.close();
}
```

Trước khi chạy chương trình để kiểm tra lỗi, trong tập tin Program.cpp ta bổ sung thêm thư viện <fstream>.

```
//Hàm xuất dữ liệu
void Output(int a[MAX], int n)
{
	int i;
	for (i = 0; i<n; i++)
		cout << a[i] << "\t";
}
//Cài đạt hàm hoán vị
void HoanVi(int &a, int &b)
{
	int tam = a;
	a = b;
	b = tam;
}
//hàm cài đặt thuật giải chọn trực tiếp
//Tai moi buoc dua gia tri nho nhat ve dau mang
void Selection_L(int a[MAX], int n)
{
	int i, j, cs_min;
	for (i = 0; i<n - 1; i++)
	{
		cs_min = i;
		for (j = i + 1; j<n; j++)
		if (a[j]<a[cs_min])
			cs_min = j;
		HoanVi(a[i], a[cs_min]);
```

```
	}
}
```

- Trong **menu.h** sửa lại hàm **XuLuMenu**.
Do các thuật giải đều thực hiện cùng một bộ dữ liệu, nên ta bổ sung vào đối của hàm XuLyMenu them 2 tham số lả mảng 1 chiều a và kích thước n., và trong case 1 xử lý chức năng chọn trực tiếp

Hàm XuLyMenu có thể sửa lại như sau :

```
void XuLyMenu(int menu, int a[MAX], int n)
{
	switch (menu)
	{
	case 0:
		system("CLS");
		cout << "\n0. Thoat khoi chuong trinh\n";
		break;
	case 1:
		system("CLS");
		cout << "\n1. Chon Truc tiep - tai moi buoc dua GTNN ve dau mang";
		cout << "\nMang ban dau:\n";
		Output(a, n);
		cout << endl;
		Selection_L(a, n);
		cout << "\nDay da sap tang:\n";
		Output(a, n);
		cout << endl;
		break;
	//…
	}
	system("PAUSE");
}
```

- Trong tập tin Program.cpp, sửa lại hàm ChayChuongTrinh:

```
void ChayChuongTrinh()
{
	int soMenu = 10,
		menu;
	int a[MAX], n = 0;
	char filename[MAX];
	cout << "\nNhap ten tap tin, filename = ";
	cin >> filename;
	File_Array(filename, a, n);
	do
	{
		system("CLS");
		menu = ChonMenu(soMenu);
		XuLyMenu(menu, a, n);
	} while (menu > 0);
}
```

//Khi chạy tiếp, chương trình đòi tập tin dữ liệu, nên ta chuẩn bị trước dữ liệu
Có thể chuẩn bị bằng cách : Nhấn chuột phải vào Source Files, chọn Add, chọn New Item, chọn tiếp Utility, chọn tiếp Text file (*.txt), đặt tên file là text1.txt và soạn thảo nội dung theo yêu cầu bài toán.

Bước 6 : Hoàn chỉnh project
	- Trong tập tin thuvien.h, cài đặt bổ sung các hàm chức năng theo yêu cầu bài toán.

- Trong tập tin menu.h, bổ sung các câu lệnh vào các case 2,3,4,5,6,9,10 để sử dụng các chức năng tương ứng.

Lưu ý là, khi thực hiện xong 1 thuật giải, dãy đã được sắp, kết quả vẫn giữ lại tính tăng đó, nên đầu vào cho lần sử dụng kế tiếp cho một thuật giải không phải là dãy ban đầu. Vậy mỗi lần thực hiện thuật giải, để lúc nào cũng có đầu vào như ban đầu, ta sẽ thực hiện lên bản sao của dãy đầu vào.

Bổ sung các nội dung sau vào project :

- Bổ sung các nội dung sau vào tập tin **thư viện.h** :

```
//Copy a sang b
void Copy(int b[MAX], int a[MAX], int n)
{
	for (int i = 0; i < n; i++)
		b[i] = a[i];
}

//Tai moi buoc, chen PT hien hanh vao mang con ben trai tang dan
void Insertion_L(int a[MAX], int n)
{
	int i, x, pos;
	for (i = 1; i<n; i++)
	{
		x = a[i];
		for (pos = i - 1; (pos >= 0) && (a[pos]>x); pos--)
			a[pos + 1] = a[pos];
		a[pos + 1] = x;
	}
}

// Doi cho truc tiep :Tai moi buoc dua gia tri nho nhat ve dau mang
void Interchange_L(int a[MAX], int n)
{
	int i, j;
	for (i = 0; i < n - 1; i++)
	{
		for (j = i + 1; j<n; j++)
		if (a[j]<a[i])
			HoanVi(a[i], a[j]);
	}
}

//Buble: Tai moi buoc dua GTNN ve dau mang
void Buble_L(int a[MAX], int n)
{
	int i, j;
	for (i = 0; i<n - 1; i++)
	{
		for (j = n - 1; j>i; j--)
		if (a[j]<a[j - 1])
			HoanVi(a[j - 1], a[j]);
	}
}

//Chen nhi phan
void Binary_Insertion(int a[MAX], int n)
{
	int l, r, m;
```

```
	int i, j;
	int x;
	for (i = 1; i<n; i++)
	{
		x = a[i]; l = 0; r = i - 1;
		while (l <= r)
		{
			m = (l + r) / 2;
			if (x<a[m])
				r = m - 1;
			else
				l = m + 1;
		}
		for (j = i - 1; j >= l; j--)
			a[j + 1] = a[j];
		a[l] = x;
	}
}

//Sap theo co so
void Radix(int a[MAX], int n)
{
	int max, m;
	max = a[0]; m = 0;
	int k, i, j, du, thuong;
	int b0[MAX], b1[MAX], b2[MAX], b3[MAX], b4[MAX], b5[MAX],
	    b6[MAX], b7[MAX], b8[MAX], b9[MAX];
	int p0, p1, p2, p3, p4, p5, p6, p7, p8, p9;
	//Tim max(a)
	for (i = 0; i<n; i++)
	if (a[i]>max)
		max = a[i];
	//Xã dinh so cac chu so cua max(a) : m
	while (max != 0)
	{
		max = max / 10;
		m++;
	}

	k = 0; //khoi tao chu so k = 0 : hang don vi
	while (k<m)
	{
		p0 = p1 = p2 = p3 = p4 = p5 = p6 = p7 = p8 = p9 = 0;//khoi tao chi so cua cac lo
		for (i = 0; i<n; i++)
		{
			//xac dinh chu so hang k cua a[i] : du
			thuong = a[i];
			for (j = 0; j <= k; j++)
			{
				du = thuong % 10;
				thuong = thuong / 10;
			}
			//Phan vao cac lo
			switch (du)
			{
```

```
                        case 0:b0[p0++] = a[i];
                                break;
                        case 1:b1[p1++] = a[i];
                                break;
                        case 2:b2[p2++] = a[i];
                                break;
                        case 3:b3[p3++] = a[i];
                                break;
                        case 4:b4[p4++] = a[i];
                                break;
                        case 5:b5[p5++] = a[i];
                                break;
                        case 6:b6[p6++] = a[i];
                                break;
                        case 7:b7[p7++] = a[i];
                                break;
                        case 8:b8[p8++] = a[i];
                                break;
                        case 9:b9[p9++] = a[i];
                                break;
                        }
                }//Phan xong vao cac lo khi xet hang k
                //Noi lai theo trinh tu de co day a tang theo hang k
                j = 0;
                for (i = 0; i<p0; i++)
                        a[j++] = b0[i];
                for (i = 0; i<p1; i++)
                        a[j++] = b1[i];
                for (i = 0; i<p2; i++)
                        a[j++] = b2[i];
                for (i = 0; i<p3; i++)
                        a[j++] = b3[i];
                for (i = 0; i<p4; i++)
                        a[j++] = b4[i];
                for (i = 0; i<p5; i++)
                        a[j++] = b5[i];
                for (i = 0; i<p6; i++)
```

- Hoàn thiện hàm XuLyMenu, thành :

```
void XuLyMenu(int menu, int a[MAX], int n)
{
        char filename[MAX];
        int b[MAX];
        switch (menu)
        {
        case 0:
                system("CLS");
                cout << "\n0. Thoat khoi chuong trinh\n";
                break;
        case 1:
                system("CLS");
                cout << "\n1. Chon Truc tiep - tai moi buoc dua GTNN ve dau mang";
                Copy(b, a, n);

                cout << "\nMang ban dau:\n";
```

```
            Output(b, n);
            cout << endl;
            Selection_L(b, n);
            cout << "\nDay da sap tang:\n";
            Output(b, n);
            cout << endl;
            break;
case 2:
            system("CLS");
            cout << "\n2. Chen Truc tiep - chen vao day con tang ben trai";
            Copy(b, a, n);
            cout << "\nMang ban dau:\n";
            Output(b, n);
            cout << endl;
            Insertion_L(b, n);
            cout << "\nDay da sap tang:\n";
            Output(b, n);
            cout << endl;
            break;

case 3:
            system("CLS");
            cout << "\n3. Doi cho Truc tiep - tai moi buoc dua GTNN ve dau mang";
            Copy(b, a, n);
            cout << "\nMang ban dau:\n";
            Output(b, n);
            cout << endl;
            cout << "\nDay da sap tang:\n";
            Interchange_L(b, n);
            Output(b, n);
            cout << endl;

            break;

case 4:
            system("CLS");
            cout << "\n4. Buble - tai moi buoc dua GTNN ve dau mang";
            Copy(b, a, n);
            cout << "\nMang ban dau:\n";
            Output(b, n);
            cout << endl;
            Insertion_L(b, n);
            cout << "\nDay da sap tang:\n";
            Output(b, n);
            cout << endl;
            break;

case 5:
            system("CLS");
            cout << "\n6. Chen nhi phan";
            Copy(b, a, n);
            cout << "\nMang ban dau:\n";
            Output(b, n);
            cout << endl;
            Binary_Insertion(b, n);
```

```
            cout << "\nDay da sap tang:\n";
            Output(b, n);
            cout << endl;
            break;
        case 6:
            system("CLS");
            cout << "\n6. Radix Sort";
            Copy(b, a, n);
            cout << "\nMang ban dau:\n";
            Output(b, n);
            cout << endl;
            Radix(b, n);
            cout << "\nDay da sap tang:\n";
            Output(b, n);
            cout << endl;
            break;
        case 9:
            system("CLS");
            cout << "\n9. Xem du lieu dang su dung";
            Output(a, n);
            cout << endl;
            break;
        case 10:
            system("CLS");
            cout << "\n10. Chon du lieu khac";
            cout << "\nNhap ten tap tin, filename = ";
            _flushall();
            cin >> filename;
            File_Array(filename, a, n);
            cout << "\nMang du lieu moi:\n";
            Output(a, n);
            cout << endl;
            break;
        }
        system("PAUSE");
}
```

- Nội dung tập tin Program.cpp chỉ thêm thư viện string.h

Lưu ý là mỗi khi viết xong một hàm thì chạy chương trình kiểm tra lỗi

Bài 2:

Giả sử có một danh sách sinh viên, mỗi sinh viên được lưu trữ các thông tin:

- Mã sinh viên, //chuỗi có 7 ký tự, không có ký tự trắng
- Họ của sinh viên, //chuỗi có không quá 10 ký tự
- Tên lót, // chuỗi có không quá 10 ký tự
- Tên sinh vien, //chuỗi có không quá 10 ký tự
- Năm sinh,// số nguyên dương
- Lớp, // chuỗi có 5 ký tự, không có ký tự trắng
- Điểm trung bình, //số thực từ 0 đến 10
- Tích lũy,//số nguyên từ 0 đến 50

Các chuỗi lưu trử thông tin về họ, tên lót, tên có thể gồm nhiều từ, các từ được nối với nhau bởi dấu gạch dưới.

- Tập tin dữ liệu "***text1.txt***":

Mã SV	Họ	Tên lót	Tên	Lớp	NSinh	Điểm TB	Tích Lũy
1412045	Nguyen	Tuan	Vo	CTK38	1996	6.2	30
1443210	Truong	Thi	Hoa	CTK38	1995	7.3	31
1423452	Tran	Ngoc	Ninh	CTK36	1994	8.6	32
1334432	Hoang	_	Hoa	CTK37	1995	5.2	36
1342332	Le	Thi	Lieu	CTK37	1994	4.6	37
1342032	Van	Thi	Hoa	CTK38	1994	6.6	38
1223052	Vo	Ngoc	Hoa	CTK36	1994	6.6	40
1312040	Nguyen	Van	Vu	CTK37	1996	6.2	41
1212045	Nguyen	Thi	Lieu	CTK37	1995	9.2	42
1313045	Tran	Trong	Hieu	CTK37	1993	4.1	43
1212042	Ly	Van	Hoa	CTK36	1993	5.3	44
1300432	Le	_	Vo	CTK37	1995	5.5	45

Viết chương trình tùy chọn sắp tăng danh sách nhân viên theo trường "Điểm trung bình" với các thuật giải:

1. QuickSort
2. HeapSort
3. MergSort

Tạo Project Lab04_D_Bai02 với 3 tập tin Thuvien.h, menu.h, program.cpp , hoàn thiện dần nội dung 3 tập tin này để có project hoàn chỉnh như bài 1.

Kết quả cuối cùng như sau :

- Nội dung tập tin Program.cpp:

```
#include <iostream>
#include <fstream>
#include <string.h>
#include <iomanip>


using namespace std;

#include "Thuvien.h"
#include "Menu.h"

void ChayChuongTrinh();

int main()
{
	ChayChuongTrinh();
	return 1;
}

void ChayChuongTrinh()
{
	int soMenu = MAX_MENU,
		menu,
		n = 0;;
	sinhvien a[MAX];
```

```
	char filename[MAX];
	cout << "\nFilename = ";
	cin >> filename;
	File_Array(filename, a, n);
	do
	{
		system("CLS");
		menu = ChonMenu(soMenu);
		XuLyMenu(menu, a, n, filename);
	} while (menu > 0);
}
```

- Nội dung tập tin menu.h:

```
//soan thao he thong menu

void XuatMenu();
int ChonMenu(int soMenu);
void XuLyMenu(int menu, sinhvien a[MAX], int &n);

//=======================================================

void XuatMenu()
{
	cout << "\n================ He thong chuc nang ===============";
	cout << "\n0. Thoat khoi chuong trinh";
	cout << "\n1. Quick Sort";
	cout << "\n2. Heap Sort";
	cout << "\n3. Merge Sort";
	cout << "\n4. Xem du lieu dang su dung";
	cout << "\n5. Chon du lieu khac";

}

void XuLyMenu(int menu, sinhvien a[MAX], int n, char filename[MAX])
{
	sinhvien b[MAX];
	switch (menu)
	{
	case 0:
		system("CLS");
		cout << "\n0. Thoat khoi chuong trinh\n";
		break;
	case 1:
		system("CLS");
		cout << "\n1. Quick Sort";
		Copy(b, a, n);
		cout << "\nMang ban dau:\n";
		Output_Arr(b, n);
		cout << endl;
		QuickSort(b, n);
		cout << "\nmang da sap tang theo truong dtb:\n";
		Output_Arr(b, n);
		cout << endl;
```

```
			break;
		case 2:
			system("CLS");
			cout << "\n2. Heap Sort";
			Copy(b, a, n);
			cout << "\nMang ban dau:\n";
			Output_Arr(b, n);
			cout << endl;
			HeapSort(b, n);
			cout << "\nmang da sap tang theo truong dtb:\n";
			Output_Arr(b, n);
			cout << endl;
			break;

		case 3:
			system("CLS");
			cout << "\n3. Merge Sort";
			Copy(b, a, n);
			cout << "\nMang ban dau:\n";
			Output_Arr(b, n);
			cout << endl;
			MergeSort(b, n);
			cout << "\nmang da sap tang theo truong dtb:\n";
			Output_Arr(b, n);
			cout << endl;

			break;

		case 4:
			system("CLS");
			cout << "\n4. Xem du lieu dang su dung";
			cout << "\nMang du lieu dang su dung:\n";
			Output_Arr(a, n);
			cout << endl;
			break;

		case 5:
			system("CLS");
			cout << "\n30. Chon du lieu khac";
			cout << "\nNhap ten tap tin, filename = ";
			_flushall();
			cin >> filename;
			File_Array(filename, a, n);
			cout << "\nMang du lieu moi:\n";
			Output_Arr(a, n);
			cout << endl;
			break;
		}

		system("PAUSE");
}

int ChonMenu(int soMenu)
{
		int stt;
```

```
	for (;;)
	{
		system("CLS"); //xoa man hinh
		XuatMenu();
		cout << "\nNhap mot so ( 0 <= so <= " << soMenu << " ) de chon menu, stt = ";
		cin >> stt;
		if (0 <= stt && stt <= soMenu)
			break;
	}
	return stt;
}
```

- Nội dung tập tin Thuvien.h:

```
//Sap danh sach nhan vien theo truong dtb
#define MAX 1000
#define MAX_MENU 5

struct sinhvien
{
	char maSV[8];
	char hoSV[10];
	char tenLot[10];
	char ten[10];
	char lop[6];
	int namSinh;
	double dtb;
	int	tichLuy;
};

//=============================================
void File_Array(char *filename, sinhvien a[MAX], int &n);
void Output_Arr(sinhvien a[MAX], int n);
void Output_Struct(sinhvien p);
void Heading();
//=============================================
void HoanVi(sinhvien &x, sinhvien &y);
void Copy(sinhvien b[MAX], sinhvien a[MAX], int n);
//=============================================
void QuickSort(sinhvien a[MAX], int n);
void Partition(sinhvien a[MAX], int n, int l, int r);
void Shift(sinhvien a[MAX], int l, int r);
void Create_Heap(sinhvien a[MAX], int n);
void HeapSort(sinhvien a[MAX], int n);
void MergeSort(sinhvien F[MAX], int n);
void  Merge(sinhvien F1[MAX], int h1, sinhvien F2[MAX], int h2, sinhvien F[MAX], int p);
void  Distribution(sinhvien F[MAX], int n, sinhvien F1[MAX], int &h1, sinhvien F2[MAX], int &h2, int p);
//=============================================
//void nhap_mang(int a[MAX], int n);
void Output_Arr(int a[MAX], int n);
void File_Array(char *filename, int a[MAX], int &n);
void Ouput_Radix(int a[MAX], int n, int k);
//=============================================
```

```
//Cai dat cac ham chuc nang

//Thuat giai phan hoach
void Partition(sinhvien a[MAX], int n, int l, int r)
{
	int i, j;
	sinhvien x;
	x = a[(l + r) / 2];
	i = l; j = r;
	do
	{
		while (a[i].dtb <x.dtb)
			i++;
		while (a[j].dtb>x.dtb)
			j--;
		if (i <= j)
		{
			HoanVi(a[i], a[j]);
			i++; j--;
		}
	} while (i <= j);

	if (l < j)
		Partition(a, n, l, j);
	if (i < r)
		Partition(a, n, i, r);
}

//Quick sort
void QuickSort(sinhvien a[MAX], int n)
{
	Partition(a, n, 0, n - 1);
}

void Shift(sinhvien a[MAX], int l, int r)
{
	int i, j;
	sinhvien x;
	i = l; j = 2 * i + 1;
	x = a[i];
	while (j <= r)
	{
		if (j<r)
		if (a[j].dtb<a[j + 1].dtb)
			j = j + 1;
		if (a[j].dtb <= x.dtb)
			return;
		else
		{
			a[i] = a[j];
			i = j;
			j = 2 * i + 1;
			a[i] = x;
		}
	}
```

```
}

void Create_Heap(sinhvien a[MAX], int n)
{
	int l;
	l = (n - 1) / 2;
	while (l >= 0)
	{
		Shift(a, l, n - 1);
		l = l - 1;
	}
}

void HeapSort(sinhvien a[MAX], int n)
{
	int r, i = 0;
	Create_Heap(a, n);
	r = n - 1;
	while (r>0)
	{
		i++;
		HoanVi(a[0], a[r]);
		r = r - 1;
		Shift(a, 0, r);
	}
}

void MergeSort(sinhvien F[MAX], int n)
{
	int  p = 1, h1, h2;
	sinhvien F1[MAX], F2[MAX];
	int i = 1;
	while (p < n)
	{
		Distribution(F, n, F1, h1, F2, h2, p);
		Merge(F1, h1, F2, h2, F, p);

		i++;
		p = p * 2;
	}
}
//*****************************
void  Distribution(sinhvien F[MAX], int n, sinhvien F1[MAX], int &h1, sinhvien F2[MAX], int &h2, int p)
{
	int i, k = 1, l = 0;
	h1 = 0; h2 = 0;
	do
	{
		i = 1;
		while (i <= p && l < n)
		{
			if (k == 1)
			{
				F1[h1++] = F[l];
			}
```

```
                        else
                        {
                                F2[h2++] = F[l];
                        }
                        i++;
                        l++;
                }
                k = 3 - k;
        } while (l < n);

}
//*****************************
void  Merge(sinhvien F1[MAX], int h1, sinhvien F2[MAX], int h2, sinhvien F[MAX], int p)
{
        int i1 = 0, i2 = 0, r1, r2;
        int h = 0;
        while (i1 < h1 && i2 < h2)
        {
                r1 = r2 = 1;
                while ((r1 <= p) && (r2 <= p) && i1 < h1 && i2 < h2)
                {
                        if (F1[i1].dtb <= F2[i2].dtb)
                        {

                                F[h++] = F1[i1];
                                r1++;
                                i1++;
                        }
                        else
                        {

                                F[h++] = F2[i2];
                                r2++;
                                i2++;
                        }
                }

                while (i1 < h1 && r1 <= p)
                {

                        F[h++] = F1[i1];
                        i1++; r1++;
                }
                while (i2 < h2 && r2 <= p)
                {

                        F[h++] = F2[i2];
                        i2++; r2++;
                }
        }
        while (i1 < h1)
        {

                F[h++] = F1[i1];
                i1++;
```

```
	}
	while (i2 < h2)
	{

		F[h++] = F2[i2];
		i2++;
	}

}
//=============================================
void File_Array(char *filename, sinhvien a[MAX], int &n)
{
	ifstream in(filename);
	if (!in)
	{
		cout << "\nLoi mo file !\n";
		system("PAUSE");
		return;
	}
	char maSV[8];
	char hoSV[10];
	char tenLot[10];
	char ten[10];

	char lop[6];
	int namSinh;
	double dtb;
	int	tichLuy;

	n = 0;
	in >> maSV; strcpy_s(a[n].maSV, maSV);
	in >> hoSV; strcpy_s(a[n].hoSV, hoSV);
	in >> tenLot; strcpy_s(a[n].tenLot, tenLot);
	in >> ten; strcpy_s(a[n].ten, ten);
	in >> lop; strcpy_s(a[n].lop, lop);
	in >> namSinh; a[n].namSinh = namSinh;
	in >> dtb; a[n].dtb = dtb;
	in >> tichLuy; a[n].tichLuy = tichLuy;

	while (!in.eof())
	{
		n++;
		in >> maSV; strcpy_s(a[n].maSV, maSV);
		in >> hoSV; strcpy_s(a[n].hoSV, hoSV);
		in >> tenLot; strcpy_s(a[n].tenLot, tenLot);
		in >> ten; strcpy_s(a[n].ten, ten);
		in >> lop; strcpy_s(a[n].lop, lop);
		in >> namSinh; a[n].namSinh = namSinh;
		in >> dtb; a[n].dtb = dtb;
		in >> tichLuy; a[n].tichLuy = tichLuy;
	}
	n++;
	in.close();
}
//Xuat tieu de
```

```
void Heading()
{
	cout<<"\n=========================================================\n";
	cout << setiosflags(ios::left);
	cout << setw(10) << "Ma SV"
		<< setw(12) << "Ho SV"
		<< setw(12) << "Tenlot SV"
		<< setw(12) << "Ten SV"
		<< setw(12) << "Lop"
		<< setw(6) << "NS"
		<< setw(6) << "DTB"
		<< setw(10) << "TichLuy";
	cout<<"\n=========================================================\n";
}

//Xuat 1 s1nh vien
void Output_Struct(sinhvien p)
{

	cout << setiosflags(ios::left)
		<< setw(10) << p.maSV
		<< setw(12) << p.hoSV
		<< setw(12) << p.tenLot
		<< setw(12) << p.ten
		<< setw(12) << p.lop
		<< setw(6) << p.namSinh
		<< setw(6) << setprecision(2) << p.dtb
		<< setw(10) << p.tichLuy;
}

//Xuat danh sach sinh vien
void Output_Arr(sinhvien a[MAX], int n)
{
	int i;
	Heading();
	for (i = 0; i < n; i++)
	{
		Output_Struct(a[i]);
		cout << '\n';
	}
	cout << "\n=========================================================\n";
}
void HoanVi(sinhvien &x, sinhvien &y)
{
	sinhvien t;
	t = x;
	x = y;
	y = t;
}
void Copy(sinhvien b[MAX], sinhvien a[MAX], int n)
{
	for (int i = 0; i < n; i++)
		b[i] = a[i];
}
```

Bài 3 :

1. Chọn trực tiếp (Selection Straight sort) : Selection_R(a,n)

Thuật giải tiến hành trong n-1 bước, Với mọi i = 0,…, n-2 , tìm giá trị max trong đoạn [a_0, a_{n-1-i}] , rồi đưa Max về cuối dãy $a_0,a_{i+1},\ldots,a_{n-1-i}$

2. Thuật giải chọn 2 đầu : Selection_R_L(a,n)

Với mọi bước i = 0 -> n/2 – 1, trong đoạn [a_i,a_{n-1-i}], đông thời đưa giá trị nhỏ nhất về đầu dãy và đưa giá trị lớn nhất về cuối dãy.

3. Chèn trực tiếp (Insertion Straight sort): Insertion_R(a,n)

Thuật giải tiến hành trong n-1 bước, Với mọi i = n-2 ↓0 , tìm vị trí thích hợp của dãy con tăng dần $a_{i+1},a_i,\ldots,a_{n-1}$ để chèn a[i] vào tạo ra dãy con i+1 phần tử $a_i,a_{i+1},\ldots,a_{n-1}$ tăng dần.
Tìm vị trí thích hợp này bằng thuật giải tím kiếm tuyến tính từ vị trí i+1 đến n-1.

4. Đổi chỗ trực tiếp (Interchange Straight sort): Interchange_R(a,n)

Thuật giải tiến hành trong n-1 bước, Với mọi j = n-1 ↓ 1 , tìm các phần tủ trước a_j và tạo với a_j thành một cặp nghịch thế, rồi triệt tiêu các cặp nghịch thế này.

5. Nổi bọt (Bubble sort): Buble_R(a,n)

Thuật giải tiến hành trong n-1 bước, Với mọi j = n-1 ↓ 1 , xuất phát từ đầu dãy đổi chỗ các cặp phần tử kế cận để đưa phần tử lớn nhất về đầu dãy .

6. Thuật giải Shaker sort (Cải tiến buble)

Trong khi l < r, thực hiện trên đoạn a[l,r] tuần tự 2 lượt :

- Lượt đi : từ biên phải tiến về trái, đổi chỗ 2 phần tử kế cận nếu cần thiết để đưa giá trị lớn nhất về cuối mảng, xác định lại biên trái l.
- Lượt về : từ biên trái tiến về phải, đổi chỗ 2 phần tử kế cận nếu cần thiết để đưa giá trị nhỏ nhất về đầu mảng, xác định lại biên phải r.

Tổ chức Project như D.bai 1, các hàm chức năng cài đặt như sau :

```
//1. Tai moi buoc dua gia tri lon nhat ve cuoi mang
void Selection_R(int a[MAX], int n)
{
	int i, j, cs_max;
	for (i = 0; i < n - 1; i++)
	{
		cs_max = n - 1 - i;
		for (j = n - 2 - i; j >= 0; j--)
		if (a[j] > a[cs_max])
			cs_max = j;
		HoanVi(a[n - 1 - i], a[cs_max]);

	}
}

//2. Tai moi buoc dua GTLN ve cuoi mang, dua GTNN ve dau mang
void Selection_R_L(int a[MAX], int n)
{
	int i, j, cs_min, cs_max;
	for (i = 0; i < n / 2; i++)
	{
		cs_min = i;
		cs_max = n - 1 - i;
		for (j = i; j <= n - 1 - i; j++)
```

```
		{
			if (a[j] < a[cs_min])
				cs_min = j;
			if (a[j] > a[cs_max])
				cs_max = j;
		}
		if (cs_min == n - i - 1) //?
		{
			HoanVi(a[i], a[cs_min]);
			if (cs_max != i) //?
				HoanVi(a[cs_max], a[n - i - 1]);
		}
		else
		{
			HoanVi(a[cs_max], a[n - i - 1]);
			HoanVi(a[i], a[cs_min]);
		}
	}
}

//3. Insertion_R : Tai moi buoc, chen PT hien hanh vao mang con ben phai tang dan
void Insertion_R(int a[MAX], int n)
{
	int i, x, pos;
	for (i = n - 2; i >= 0; i--)
	{
		x = a[i];
		for (pos = i + 1; (pos < n) && (a[pos] < x); pos++)
			a[pos - 1] = a[pos];
		a[pos - 1] = x;
	}
}

//4. Interchange_R: Tai moi buoc dua gia tri lon nhat ve cuoi mang
void Interchange_R(int a[MAX], int n)
{
	int i, j;
	for (j = n-1; j > 0; j--)
	{
		for (i = 0; i < j; i++)
		if (a[i] > a[j])
			HoanVi(a[i], a[j]);
	}
}

//5. Buble_R : Tai moi buoc dua GTLN ve cuoi mang
void Buble_R(int a[MAX], int n)
{
	int i, j;
	for (j = n-1; j > 0; j--)
	{
		for (i = 0; i < n-1; i++)
		if (a[i] > a[i + 1])
			HoanVi(a[i + 1], a[i]);
		cout << "\nBuoc " << n - j  << " : ";
```

```
			Output(a, n);
			cout << "\n";
		}
		cout << "\nCo " << n-1 << " buoc thuc hien thuat giai.\n";
}

//6. Shaker :
void Shaker(int a[MAX], int n)
{
		int l = 0, r = n - 1;
		int k = n - 1;
		int j;
		while (l<r)
		{
			buoc++;
			j = r;//Khoi tao j tu bien phai
			while (j>l)
			{
				if (a[j]<a[j - 1])
				{
					HoanVi(a[j], a[j - 1]);
					k = j;
					cout << "\nk = " << k;
				}
				j = j - 1;
			}
			l = k;//xac dinh lai bien trai l
			j = l; //khoi tao j tu bien trai
			while (j<r)
			{
				if (a[j]>a[j + 1])
				{
					HoanVi(a[j], a[j + 1]);
					k = j;
				}
				j = j + 1;
			}
			r = k;//xac dinh lai bien phai l
		}
}
```

E. Bài tập

Bài 1:

Viết chương trình tùy chọn sắp tăng dãy n số nguyên gồm các chức năng:

1. Thuat giai Chon Truc tiep - tai moi buoc dua GTNN ve dau mang
2. Thuat giai Chon Truc tiep - tai moi buoc dua GTLN ve cuoi mang
3. Thuat giai Chon hai dau
4. Thuat giai Chen Truc tiep - chen vao day con tang ben trai
5. Thuat giai Chen Truc tiep - chen vao day con tang ben phai
6. Thuat giai Chen nhi phan
7. Thuat giai Doi cho Truc tiep - tai moi buoc dua GTNN ve dau mang"
8. Thuat giai Doi cho Truc tiep - tai moi buoc dua GTLN ve cuoi mang
9. Thuat giai Buble - tai moi buoc dua GTNN ve dau mang";
10. Thuat giai i Buble - tai moi buoc dua GTLN ve cuoi mang";

11. Thuat giai Quick Sort";
12. Thuat giai Merge Sort
13. Thuat giai Heap Sort";
14. Thuat giai Radix Sort";
15. Thuat giai Shaker Sort";
19. Xem Du lieu hien hanh
20. Chon lai du lieu

- Tập tindữ liệu "***text1.txt*** "

8 //Kich thuoc mang

0 12 9 5 10 16 4 42

- Tập tindữ liệu "***text2.txt*** "

12 //Kich thuoc mang

113 8425 1239 428 1424 7009 4518 3252 9070 999 1725 9701

Bài 2 và Bài 3 (dùng chung dữ liệu):

Giả sử có một danh sách nhân viên, mỗi nhân viên được lưu trữ các thông tin:

- Mã nhân viên, //chuỗi có 7 ký tự, không có ký tự trắng
- Họ của nhân viên, // chuỗi có không quá 10 ký tự
- Tên lót nhân viên, // một chuỗi có không quá 10 ký tự
- Tên nhân vien, // chuỗi có không quá 10 ký tự
- Địa chỉ, // chuỗi có không quá 15 ký tự
- Năm sinh, //số nguyên dương 4 ký số
- Lương, //số thực dương

- Tập tin dữ liệu "***text.txt*** " lưu trử danh sách nhân viên :

Mã NV	Họ	Tên lót	Tên	Địa Chỉ	NSinh	Lương
LD12045	Nguyen	Tuan	Vo	Lam_Dong	1980	25000000
LD13210	Ly	Van	Hoa	Ninh_Thuan	1985	30000000
LD13452	Tran	Ngoc	Ninh	Khanh_Hoa	1974	10000000
LD14432	Nguyen	_	Vo	Phu_Yen	1985	12000000
LD15332	Le	Thi	Lieu	Binh_Dinh	1974	10000000
LD22032	Van	Thi	Hoa	Lam_Dong	1984	10000000
LD22052	Vo_Hoang	Ngoc	Hoa	Lam_Dong	1984	70000000
LD22140	Tran	Vuong	Vo	Binh_Dinh	1990	12000000
LD22145	Le	Thi	Vo	Khanh_Hoa	1986	12000000
LD23045	Tran_Nguyen	Trong	Hieu	Ha_Noi	1991	25000000
LD24042	Ly	Van	Hoa	Ha_Noi	1983	30000000
LD30432	Nguyen	_	Vo	Lam_Dong	1975	12000000

Bài 2:

Viét chương trình tùy chọn sắp danh sách nhân viên theo yêu cầu :

- Dùng thuật giải chọn trực tiếp sắp danh sách tang dần theo mã NV.
- Dùng thuật giải chèn trực tiếp sắp danh sách tang dần theo địa chỉ.
- Dùng thuật giải Radix sắp danh sách tăng dần theo năm sinh

Bài 3:

Sắp danh sách nhân viên theo yêu cầu :

- Sắp tăng dần theo mức lương , dùng một trong các thuật giải :

    - Quick Sort,
    - Heap Sort,
    - Merge Sort
    - Shaker Sort
- Nếu cùng mức lương, sắp tăng dần theo tên
- Nếu cùng lương, tên, sắp tăng dần theo họ
- Nếu cùng lương, tên, họ, sắp tăng dần theo năm sinh..

Bài 4 :

Viết chương trình tùy chọn đếm số lượng các phép toán so sánh, các phép hoán vị trong các thuật giải :

1. Thuat giai Chon Truc tiep
2. Thuat giai Chen Truc tiep
3. Thuat giai Doi cho Truc tiep"
4. Thuat giai Buble

Với tập tin dữ liệu : *"text.txt "*

84 //Kich thuoc mang

0	12	9	5	10	16	4	42	10	0	9	12
9	0	6	5	21	23	26	54	32	21	10	0
0	9	8	7	8	9	0	4	3	6	2	1
9	0	9	4	6	2	7	6	1	8	8	0
9	10	9	14	16	12	7	6	1	18	8	10
20	19	28	7	8	9	10	14	3	6	12	21
10	2	9	15	1	6	4	4	1	0	9	2