

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

-----*-----

BÁO CÁO ĐỒ ÁN MÔN HỌC

CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

Giảng viên hướng dẫn : ĐỖ VĂN NHƠN

NGUYỄN BÍCH VÂN

Lớp : IT003.H21.CLC

Sinh viên thực hiện : Nguyễn Thành Luân - 16520703

Mục lục

BÁO CÁO ĐỒ ÁN MÔN HỌC.....	1
CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT.....	1
1. Cấu trúc bài nộp	3
2. Tóm tắt đề bài	3
3. Đặc tả kĩ thuật	3
a. Các thành phần chia mã nguồn.....	3
b. Các kĩ thuật, thư viện, công nghệ đã sử dụng.....	5
4. Các tính năng của chương trình.....	6
a. Tính năng sinh mảng input ngẫu nhiên.....	6
b. Tính năng so sánh thời gian xử lí của 6 thuật giải	7
c. Tính năng xuất kết quả sau khi sắp xếp	8
5. Kết luận.....	9

1. Cấu trúc bài nộp

Bài nộp gồm có 3 thư mục là Release, Documents và Source.

- Release
 - Chứa toàn bộ chương trình và các tập tin dữ liệu đi kèm.
- Documents
 - Chứa báo cáo đồ án
- Source
 - Chứa mã nguồn chương trình.

2. Tóm tắt đề bài

Đồ án 1:

Xây dựng chương trình kiểm tra, minh họa, đánh giá thử nghiệm các thuật giải sắp xếp dãy: 3 thuật giải $O(n^2)$, 3 thuật giải $O(n \log n)$.

3. Đặc tả kĩ thuật

a. Các thành phần chia mã nguồn

- main.cpp
 - Vòng lặp chính để chạy chương trình, thể hiện dưới dạng menu cho phép người dùng lựa chọn tính năng.

```

while(true) {
    system("cls");
    setTextColor(15);

    cout<< "-----| MENU |-----\n";
    cout<< "1. Array Generation\n";
    cout<< "2. Sorting Comparison\n";
    cout<< "3. Print Result\n";
    cout<< "4. Exit\n\n";
    cout<< "--> "; cin >> keyIn;

    switch(keyIn) {
        case 1: ArrayGenerator(); break;
        case 2: showCompareResult(); break;
        case 3: printOutput(); break;
        case 4: return 0;
        default: break;
    }
}

```

- ArrayGenerator.h
 - File header chứa đoạn chương trình phát sinh ra mảng ngẫu nhiên với các thông số do người dùng tùy chọn (số lượng phần tử, phần tử nhỏ nhất, phần tử lớn nhất)
- SortAlgorithm.h và SortAlgorithm.cpp
 - Source code chứa 6 loại thuật giải sắp xếp dãy, bao gồm:
 - 3 thuật giải có độ phức tạp $O(n^2)$: Bubble Sort, Selection Sort và Insertion Sort.
 - 3 thuật giải có độ phức tạp $O(n \log n)$: Quick Sort, Heap Sort và Merge Sort.
 - **Tất cả các thuật giải sắp xếp sẽ đưa dãy về thứ tự tăng dần.**
- Function.h và Function.cpp
 - Chứa tất cả các hàm được sử dụng cho việc chạy chương trình, gồm có:
 - **getInput() : void** → Đọc dữ liệu từ các file input có sẵn (file input được tạo ra bằng chương trình phát sinh mảng ngẫu nhiên, bao gồm 3 file Text Document tương ứng với 3 loại test case : Random, Ascending, Descending).
 - **SortType(long long index, vector<long long>& elements) : void**
→ Hàm gọi đến các thuật toán sắp xếp dãy, căn cứ vào chỉ số index để xác định thuật giải gọi đến.

```

void SortType(long long index, vector<long long>& elements) {
    switch(index) {
        case 0: BubbleSort(elements); break;
        case 1: SelectionSort(elements); break;
        case 2: InsertionSort(elements); break;
        case 3: QuickSort(elements, 0, elements.size() - 1); break;
        case 4: HeapSort(elements); break;
        case 5: MergeSort(elements, 0, elements.size() - 1); break;
    }
}

```

- **getTimeHandle() : long long**
- **showTimeHandleAll() : void**
- **showCompareResult() : void**

-> Lấy thời gian cần để xử lý của một giải thuật sắp xếp, biểu diễn tất cả các kết quả thu được dưới dạng bảng so sánh thời gian tương ứng với từng loại input.

Tư tưởng : Lấy dữ liệu → Bắt đầu việc tính thời gian → Thực hiện việc sắp xếp (bảng 1 trong 6 thuật giải) → Kết thúc việc tính thời gian → Thời gian xử lý = Thời gian kết thúc – Thời gian bắt đầu.

- **printOutput() : void** -> Xuất kết quả dãy thu được ra file sau khi thực hiện 1 trong 6 thuật giải sắp xếp → dùng để kiểm tra kết quả, đối chiếu độ chính xác sau này.
- Ngoài ra còn một số hàm phụ dùng để thực hiện việc biểu diễn kết quả trên màn hình console ...

b. Các kĩ thuật, thư viện, công nghệ đã sử dụng

- Kĩ thuật sinh số ngẫu nhiên : rand() (thuộc thư viện <stdlib.h>).
- Kĩ thuật đọc, xuất file Text Document.
- 6 loại thuật giải sắp xếp dãy.
- Kĩ thuật chia file mã nguồn (.h và .cpp) để dễ dàng quản lí.
- Kĩ thuật sử dụng kiểu dữ liệu vector để lưu trữ dữ liệu với số lượng lớn và truy xuất dễ dàng.
- High – Resolution Timer : Đo thời gian xử lý chương trình với độ chính xác cao (đơn vị nano giây = $1/10^9$ giây).
- Source code được build trên phần mềm Code::Blocks.

4. Các tính năng của chương trình

a. Tính năng sinh mảng input ngẫu nhiên

- Chương trình sinh ra mảng input theo 3 trường hợp : Ngẫu nhiên, Tăng dần, Giảm dần và lưu các kết quả vào 3 file .txt (arrayRandom.txt, arrayAscending.txt, arrayDescending.txt)
- Cách dùng:
 - Tại giao diện Menu của chương trình, nhập 1 và nhấn phím Enter để sử dụng tính năng tạo mảng input.

```
-----| MENU |-----
1. Array Generation
2. Sorting Comparison
3. Print Result
4. Exit

--> 1
```

- Tại giao diện của chức năng sinh mảng input, nhập vào đúng 3 thông số theo thứ tự lần lượt là :
<số lượng phần tử>(khoảng cách)<phần tử nhỏ nhất>(khoảng cách)<phần tử lớn nhất>
và nhấn Enter để quá trình sinh mảng được bắt đầu.

```
--- ARRAY GENERATOR ---

USAGE: <number_of_elements> <minimum_value> <maximum_value>

--> 10000 0 9999
```

(trường hợp này sinh ra input gồm 10000 phần tử nằm trong khoảng [0, 9999])

- Quá trình sinh mảng sẽ diễn ra trong chốc lát và sẽ có thông báo khi hoàn thành. Khi đó nhấn Enter để trở về giao diện Menu ban đầu.

```
--- ARRAY GENERATOR ---

USAGE: <number_of_elements> <minimum_value> <maximum_value>

--> 10000 0 9999

Processing ...

OUTPUT SUCCESSFULLY :)
```

b. Tính năng so sánh thời gian xử lý của 6 thuật giải

- So sánh thời gian xử lý của 6 loại thuật giải sắp xếp dãy tương ứng với 3 loại input, thể hiện trực quan dưới dạng bảng so sánh.
- Cách dùng:
 - Tại giao diện Menu của chương trình, nhập 2 và nhấn phím Enter để sử dụng tính năng so sánh các loại giải thuật.

```
-----| MENU |-----
1. Array Generation
2. Sorting Comparison
3. Print Result
4. Exit
--> 2
```

- Quá trình tính toán và xử lý kết quả sẽ mất khá thời gian bởi vì số lượng phần tử rất lớn và việc lặp đi lặp lại các công việc : (Lấy dữ liệu → Sử dụng thuật giải sắp xếp trên 3 loại dữ liệu → Xuất).

Test case: 10000 elements
Time unit: nanosecond = $10^{(-9)}$ second

Processing...

	Random Input	Ascending Input	Descending Input
Bubble Sort	1001170000	451298000	1255828000
Selection Sort			
Insertion Sort			
Quick Sort			
Heap Sort			
Merge Sort			

- Và đây là kết quả của tính năng so sánh này :

Test case: 10000 elements
Time unit: nanosecond = $10^{(-9)}$ second

DONE !!!

	Random Input	Ascending Input	Descending Input
Bubble Sort	1005172000	451298000	1255837000
Selection Sort	547864000	634434000	681952000
Insertion Sort	316206000	501000	646929000
Quick Sort	2501000	1501000	1001000
Heap Sort	4503000	4503000	3503000
Merge Sort	3005000	1996000	2999000

RED - Worst Result

GREEN - Best Result

- Nhấn Enter để trở về giao diện Menu ban đầu.

c. Tính năng xuất kết quả sau khi sắp xếp

- Để đơn giản cho việc kiểm tra tính chính xác của các giải thuật (liệu thuật giải sắp xếp có đưa dãy ban đầu về dạng sắp xếp tăng dần hay không), tính năng này sinh ra để người dùng có thể trực tiếp xem xét việc này thông qua file outputResult.txt .
- Cách dùng:
 - Tại giao diện Menu của chương trình, nhập 3 và nhấn phím Enter để sử dụng tính năng xuất kết quả.

```

-----| MENU |-----
1. Array Generation
2. Sorting Comparison
3. Print Result
4. Exit

--> 3

```


- Tiếp theo nhập vào số thứ tự của loại thuật giải sắp xếp mà ta muốn xem kết quả, sau đó nhấn Enter. Quá trình này diễn ra trong chốc lát và sẽ có thông báo khi hoàn thành. Khi đó nhấn Enter để quay trở lại giao diện Menu ban đầu.

```
Choose one Sort-Algo that you want to get the output result:
1. Bubble Sort
2. Selection Sort
3. Insertion Sort
4. Quick Sort
5. Heap Sort
6. Merge Sort
--> 2
DONE !!!
```

5. Kết luận

Đồ án này được thực hiện trong quá trình học quân sự, nên có thể độ hoàn thiện và các tính năng của chương trình vẫn chưa đạt được một số yêu cầu nhất định đề ra ban đầu. Em mong thầy / cô thông cảm và vô cùng biết ơn thầy / cô trong suốt khoảng thời gian vừa qua đã mang đến cho em những kiến thức cần thiết, bổ ích để ứng dụng trong đồ án này nói riêng và cả công việc về sau nói chung!

Vì thời gian có hạn nên em không thể làm thêm tính năng minh họa cách chạy từng bước của các loại thuật toán sắp xếp (thực hiện trên chế độ đồ họa, sử dụng thư viện graphics.h). Nếu có thời gian, chắc chắn em sẽ tiếp tục bổ sung và hoàn thiện thêm nhiều tính năng hơn nữa cho đồ án của mình!

Một lần nữa, em xin chân thành cảm ơn thầy / cô và chúc thầy / cô luôn mạnh khỏe, hạnh phúc!