

CÁC THUẬT TOÁN SẮP XẾP

Trong bài này, ta xét bài toán sắp xếp tăng dần.

1 Yêu cầu

1.1 Tóm tắt sơ bộ

- Các bạn sẽ cài đặt các thuật toán: Selection Sort, Insertion Sort, Binary Insertion Sort, Bubble Sort, Shake Sort, Shell Sort, Heap Sort, Merge Sort, Quick Sort, Counting Sort, Radix Sort, Flash Sort.
- Sau khi cài đặt xong 7 thuật toán, các bạn sẽ tiến hành các thí nghiệm để có thể thấy được thời gian thực thi của mỗi thuật toán ứng với mỗi tình trạng dữ liệu đầu vào cũng như là so sánh thời gian thực thi của các thuật toán với nhau. Ở đây, ta xét 4 trình trạng dữ liệu đầu vào:
 - Đã có thứ tự.
 - Có thứ tự ngược.
 - Ngẫu nhiên.
 - Gần như có thứ tự.

1.2 Cụ thể

Nội dung nộp của các bạn gồm có 2 phần: phần code (7đ) và phần báo cáo (3đ).

1.2.1 Phần code (7 điểm)

Ta sẽ thí nghiệm với 4 tình trạng dữ liệu: đã có thứ tự (sorted), có thứ tự ngược (reversed), ngẫu nhiên (random), và gần như có thứ tự (nearly sorted). Với mỗi tình trạng, ta khảo sát 5 kích thước dữ liệu (số phần tử) khác nhau: 1000, 3000, 10000, 30000, 100000.

Với mỗi tình trạng dữ liệu **state**,
với mỗi kích thước dữ liệu **size**,
với mỗi thuật toán sắp xếp **sortAlg**, chương trình của các bạn cần phải:

1. Tạo mảng dữ liệu có tình trạng **state** và có kích thước **size**.
2. Sắp xếp mảng dữ liệu này bằng thuật toán **sortAlg** và đo thời gian thực thi của **sortAlg**.
3. Xuất kết quả sắp xếp ra file:
 - Mỗi phần tử cách nhau khoảng trắng.
 - Tên file đặt theo định dạng: **<sortAlg>_<state>_<size>.txt**
Ví dụ: **QuickSort_NearlySorted_1000.txt**

Các file kết quả: ngoài các file lưu kết quả sắp xếp như đã nói ở trên, để tiện lợi cho việc thống kê kết quả, chương trình của các bạn cũng cần phải xuất thời gian thực thi của tất cả các lần chạy ở trên ra một file chung:

- Ta qui ước đặt tên file này là **Result.csv**. Định dạng ***.csv** có thể mở bằng Excel, sẽ dễ dàng cho việc vẽ đồ thị.
- File **Result.csv** sẽ có dạng như hình 1. Trong đó, ở mỗi dòng, ô đầu tiên là tình trạng dữ liệu ban đầu, ô thứ hai là kích thước dữ liệu, các ô tiếp theo là thời gian thực thi của các thuật toán (mình mới chỉ điền đại bằng tay hai dòng đầu tiên để minh họa, chương trình của các bạn cần điền hết tất cả các ô thời gian thực thi).

	A	B	C	D	E	F	G	H	I
1	Input State	Input Size	Bubble	Shaker	Selection	Insertion	Quick	RandomizedQuick	Merge
2	Random	1000	0	0	0	0	0	0	0
3	Random	3000	0.1	0.1	0.1	0.1	0	0	0
4	Random	10000							
5	Random	30000							
6	Random	100000							
7	Sorted	1000							
8	Sorted	3000							
9	Sorted	10000							
10	Sorted	30000							
11	Sorted	100000							

Hình 1: Minh họa file Result.csv

- Ghi kết quả ra file *.csv như thế nào? Nếu ta mở file *.csv ở trên bằng Notepad++ thì sẽ thấy như hình 2. Như vậy, thực chất mỗi ô trên một dòng cách nhau bởi dấu phẩy. Ví dụ, ta có thể ghi dòng thứ 2 bằng câu lệnh:

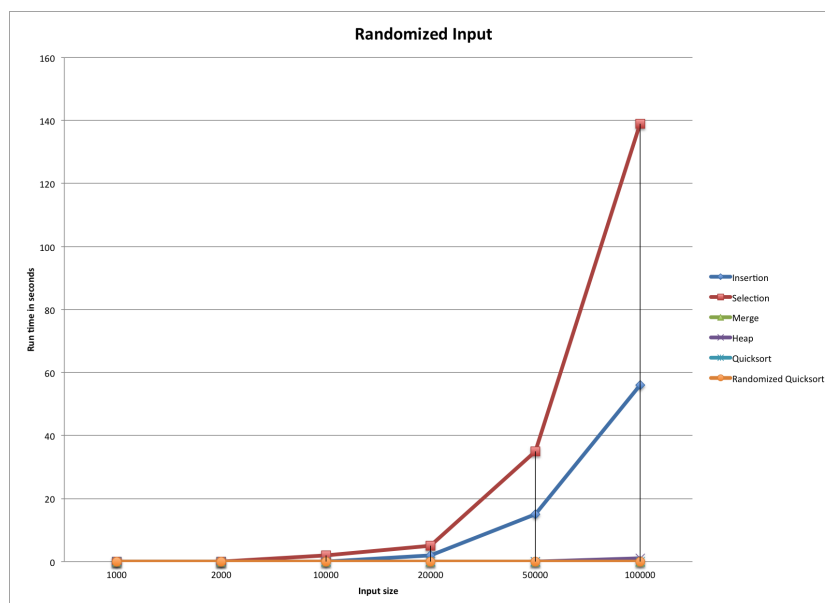
```
1 fprintf(f, "Random,1000,0,0,0,0,0,0,0\n");
```

1	Input State	Input Size	Bubble	Shaker	Selection	Insertion	Quick	RandomizedQuick	Merge
2	Random	1000	0	0	0	0	0	0	0
3	Random	3000	0.1	0.1	0.1	0.1	0	0	0

Hình 2: File Result.csv mở bằng Notepad++

1.2.2 Phần báo cáo (3 điểm)

Trình bày về kết quả thí nghiệm (1 điểm) và cho nhận xét (3 điểm). Cách trình bày kết quả thí nghiệm: thay vì đưa ra các con số về thời gian chạy, ta nên trực quan hóa bằng đồ thị; như vậy, sẽ dễ quan sát và nhận xét hơn. Các bạn sẽ vẽ 4 đồ thị ứng với 4 tình trạng dữ liệu đầu vào. Trong đó, mỗi đồ thị có trục tung là kích thước dữ liệu, trục hoành là thời gian chạy (có dạng như hình 3).



Hình 3: Minh họa dạng đồ thị

2 Một số qui định

- Cấu trúc thư mục nộp bài: trong thư mục <MSSV> gồm có:
 1. Thư mục Code: chứa toàn bộ tập tin source code (*.h, *.cpp).
 2. Thư mục Result: chứa toàn bộ các file kết quả.
 3. File Report.doc(x) hoặc Report.pdf.
- *Lưu ý: Những trường hợp giống bài nhau sẽ bị 0 điểm.*