

ĐỒ ÁN 1

CÁC THUẬT TOÁN SẮP XẾP

A- Qui định thực hiện đồ án

- Thời gian thực hiện: **3 tuần, 14/03/2016 – 27/03/2016**
- Đây là **đồ án cá nhân**.
- Source code được cài đặt bằng **ngôn ngữ C#**, có thể chạy được trên môi trường Visual Studio for Windows từ phiên bản 2012 trở lên.
- Cách thức nộp bài: nộp qua link Dropbox do giáo viên cung cấp.
- Tổ chức bài nộp
 - Thư mục CODE: chứa toàn bộ source code của chương trình, giữ nguyên cấu trúc thư mục bài làm (các tập tin solution, project), chỉ bỏ thư mục Debug và Release
 - Thư mục DOC: chứa báo cáo đồ án, lưu ở định dạng *.pdf. Báo cáo ghi hướng dẫn sử dụng chương trình và mô tả chi tiết hàm cài đặt cho từng yêu cầu của đồ án.
 - Đặt tất cả các thư mục con vào một thư mục chung, đặt tên là <MSSV hoặc CMND>_<Họ và tên> (thay nội dung < > bằng thông tin cá nhân của mình) rồi nén theo định dạng *.zip hoặc *.rar.
 - **Sinh viên không cần tạo tập tin *.exe trước (để nhiễm virus)**, giáo viên khi chấm sẽ build lại chương trình để kiểm tra source code.
- Quy định về nguồn tham khảo
 - Khuyến khích sinh viên tham khảo nhiều nguồn tài liệu để hoàn thiện bài làm nhưng sinh viên phải ghi rõ các nguồn tham khảo vào báo cáo và không được phép sao chép các source code có sẵn.
 - Sinh viên không được sao chép hoặc chia sẻ source code với những sinh viên khác cũng đang thực hiện đồ án.
 - Các bài làm vi phạm những quy định nói trên sẽ bị 0 điểm.

B- Dữ liệu đầu vào và dữ liệu đầu ra

Tập tin dữ liệu đầu vào chứa dãy số gồm N số nguyên không âm có miền giá trị thuộc [0, 99999].

- Dòng thứ nhất: số phần tử trong dãy số – N
- Dòng thứ hai: các phần tử của dãy số, hai phần tử cách nhau bằng một khoảng trắng, giá trị của chúng có thể trùng nhau

Ví dụ:

Input.txt	Giải thích
10000	Số lượng phần tử
100 2901 3333 100 567 7899 9876 3456.....78	Các phần tử của dãy số

Đề án có các tập tin đầu vào như sau:

- input_10000.txt chứa dãy số gồm 10,000 phần tử có thứ tự ngẫu nhiên
- input_100000.txt chứa dãy số gồm 100,000 phần tử có thứ tự ngẫu nhiên
- input_500000.txt chứa dãy số gồm 500,000 phần tử có thứ tự ngẫu nhiên
- input_random.txt chứa dãy số gồm 10,000 phần tử có thứ tự ngẫu nhiên (tương tự như input_10000.txt)
- input_same.txt chứa dãy số gồm 10,000 phần tử giống nhau
- input_ascending.txt chứa dãy số gồm 10,000 phần tử có thứ tự tăng dần
- input_descending.txt chứa dãy số gồm 10,000 phần tử có thứ tự giảm dần

Tập tin đầu ra chứa dãy số gồm N số nguyên không âm từ tập tin đầu vào nhưng **đã được sắp xếp**.

- Dòng thứ nhất: số phần tử trong dãy số – N
- Dòng thứ hai: các phần tử của dãy số, hai phần tử cách nhau bằng một khoảng trắng

Output.txt	Giải thích
10000	Số lượng phần tử
78 100 100 567 2901 3333 3456 7899 9876.....	Các phần tử của dãy số đã sắp xếp

C- Yêu cầu thực hiện

Yêu cầu 1 đến 6 làm việc trên tập dữ liệu đầu vào **input_10000.txt**.

- [1.0 điểm]** Cho biết trạng thái ban đầu của dãy số trong tập dữ liệu đầu vào
 - Nếu mọi phần tử trong dãy số bằng nhau, in ra màn hình dòng chữ “Day so giống nhau”
 - Nếu dãy số tăng dần ($a_i \leq a_{i+1}$ và $\exists i, j \ a_i \neq a_j$), in ra màn hình dòng chữ “Day so tang dan”
 - Nếu dãy số giảm dần ($a_i \geq a_{i+1}$ và $\exists i, j \ a_i \neq a_j$), in ra màn hình dòng chữ “Day so giam dan”
 - Các trường hợp khác, in ra màn hình dòng chữ “Day so ngau nhien”
- [1.0 điểm]** Giả sử rằng thao tác đổi chỗ hai phần tử a và b được thực hiện bằng cách gọi hàm Swap như bên dưới


```
static void Swap (ref int a, ref int b){
    int temp = a;
    a = b;
    b = temp;
}
```

Áp dụng thuật toán **sắp xếp chọn** để sắp xếp dãy số tăng dần và ghi kết quả vào tập tin đầu ra **selectionsort.txt** theo định dạng đã mô tả ở mục A-Dữ liệu đầu vào và đầu ra.
In ra màn hình số lần đổi chỗ hai phần tử a và b trong thuật toán (tức là số lần gọi hàm Swap)
- [1.5 điểm]** Thực hiện tương tự Yêu cầu 2 nhưng áp dụng thuật toán **sắp xếp vun đống** và tập tin đầu ra là **heapsort.txt**. Cũng in ra màn hình số lần gọi hàm Swap.
- [1.5 điểm]** Thực hiện tương tự Yêu cầu 2 nhưng áp dụng thuật toán **sắp xếp nhanh** và tập tin đầu ra là **quicksort.txt**. Cũng in ra màn hình số lần gọi hàm Swap.
- [1.5 điểm]** Thực hiện tương tự Yêu cầu 2 nhưng áp dụng thuật toán **sắp xếp trộn** và tập tin đầu ra là **mergesort.txt**. Cũng in ra màn hình số lần gọi hàm Swap.
- [1.5 điểm]** Áp dụng thuật toán **sắp xếp theo cơ số** để sắp xếp dãy số tăng dần. Ghi kết quả vào tập tin **radixsort.txt** theo định dạng đã mô tả ở mục A-Dữ liệu đầu vào và đầu ra. Không cần in ra màn hình số lần gọi hàm Swap vì trong thuật toán không có thao tác này.
- [1.0 điểm]** Lần lượt chạy 4 thuật toán sắp xếp trên với các tập dữ liệu đầu vào **input_10000.txt**, **input_100000.txt** và **input_500000.txt** để lấy thông tin về số lần gọi hàm Swap và điền vào bảng bên dưới (cột biểu diễn thuật toán, dòng biểu diễn số phần tử trong dãy số). Dựa vào sự biến thiên của số lần gọi hàm Swap, hãy so sánh hiệu quả của các thuật toán khi làm việc với dãy số có kích thước tăng dần.

	Sắp xếp chọn	Sắp xếp vun đống	Sắp xếp nhanh	Sắp xếp trộn
10,000 phần tử				
100,000 phần tử				
500,000 phần tử				

8. **[1.0 điểm]** Lần lượt chạy 4 thuật toán sắp xếp trên với các tập dữ liệu đầu vào **input_same.txt**, **input_random.txt**, **input_ascending.txt** và **input_descending.txt** để lấy thông tin về số lần gọi hàm Swap và điền vào bảng bên dưới (cột biểu diễn thuật toán, dòng biểu diễn tình trạng ban đầu của dãy số). Dựa vào sự biến thiên của số lần gọi hàm Swap, hãy so sánh hiệu quả của các thuật toán khi làm việc với dãy số có tình trạng ban đầu khác nhau.

	Sắp xếp chọn	Sắp xếp vun đống	Sắp xếp nhanh	Sắp xếp trộn
Dãy số giống nhau				
Dãy số ngẫu nhiên				
Dãy số tăng dần				
Dãy số giảm dần				

9. **[+1.0 điểm]** Hiệu chỉnh thuật toán **sắp xếp vun đống** để sắp xếp dãy số giảm dần với **tập dữ liệu đầu vào input_10000.txt**. Ghi kết quả sắp xếp vào tập tin **heapsort_descending.txt** và in ra màn hình số lần gọi hàm Swap giống như yêu cầu 2. Lưu ý: phải hiệu chỉnh một số bước trong thuật toán chứ không phải hiệu chỉnh trên kết quả sắp xếp (ví dụ: sắp mảng tăng dần rồi đảo ngược các phần tử lại → không hợp lệ).
10. **[+1.0 điểm]** Khử đệ quy thuật toán sắp xếp nhanh. So sánh thời gian chạy (tính theo milliseconds) của phiên bản hiệu chỉnh với phiên bản gốc trên **tập dữ liệu đầu vào input_same.txt**, **input_random.txt**, **input_ascending.txt** và **input_descending.txt**.

	Phiên bản đệ quy	Phiên bản khử đệ quy
Dãy số giống nhau		
Dãy số ngẫu nhiên		
Dãy số tăng dần		
Dãy số giảm dần		

LƯU Ý:

- Sinh viên không được sử dụng hàm sắp xếp cung cấp sẵn trong các thư viện.
- Dãy số đầu vào phải được đọc tự động từ tập tin đầu vào, nếu chương trình yêu cầu nhập dãy số từ bàn phím thì sẽ không được chấm tiếp.
- Đảm bảo tập tin kết quả đúng định dạng vì các tập tin này sẽ được kiểm tra tự động. Nếu không kiểm tra được thì bài làm không có điểm.