

# Cấu Trúc Dữ Liệu và Giải Thuật

## Bài Tập Thực Hành Tuần #1

06/10/2015

**Deadline: 23h59 ngày 21/10/2015**

Một số lưu ý quan trọng:

- Bài tập thực hành tuần #1 gồm 12 bài tập nhỏ: các thuật toán tìm kiếm và sắp xếp.
- Mỗi bài được kiểm thử với 5 bộ test (0.15 điểm/bộ test đúng). 1 điểm còn lại được tính dựa trên cách trình bày code. Ngoài ra, sinh viên có thể viết báo cáo để lấy điểm cộng (tối đa +1 điểm) cho từng bài thực hành.
- Cấu trúc chương trình gồm các file sau (MSSV: mã số sinh viên):
  - MSSV.cpp: chứa hàm main.
  - MSSV\_SortAls.cpp: chứa nội dung cài đặt các thuật toán sắp xếp.
  - MSSV\_SortAls.h: chứa prototype các hàm sắp xếp.
  - MSSV\_SearchAls.cpp: chứa nội dung cài đặt các thuật toán tìm kiếm.
  - MSSV\_SearchAls.h: chứa prototype các hàm tìm kiếm.
- Hàm main đọc dữ liệu từ file text tên data.txt có dạng:

```
[ext1test1]
<data1.1>
[ext1test2]
<data1.2>
...
[ext2test1]
<data2.1>
...
[ext12test5]
<data12.5>
```

Trong đó, [ext1test2] đánh dấu dòng tiếp theo sau đó trong file data.txt là dữ liệu bộ test thứ 2 của bài tập 1, <data1.2> là dữ liệu và có dạng như các ví dụ sau:

3 0 1 0:1	Dữ liệu dành cho các thuật toán tìm kiếm. Số sau dấu ":" là giá trị cần tìm trong dãy số trước dấu ":". Ta chỉ thực hiện trên các số nguyên.
3 0 1 0	Dữ liệu dành cho các thuật toán sắp xếp gồm dãy các số không âm và cách nhau bởi duy nhất 01 khoảng trắng.

Vậy ta có tất cả 60 bộ test cho bài thực hành tuần #1.

- Hàm main gọi các hàm tìm kiếm, sắp xếp và thực hiện trên bộ dữ liệu tương ứng.
- Sau khi thực hiện các hàm tìm kiếm và sắp xếp, hàm main xuất kết quả ra file output\MSSV.txt. Trong đó, output là thư mục nằm cùng với các tập tin mã nguồn và MSSV là mã số sinh viên. MSSV.txt có dạng như sau:

```
[ext1test1]
<output1.1>
[ext1test2]
<output1.2>
...
[ext2test1]
<output2.1>
...
[ext12test5]
<output12.5>
```

Trong đó, [ext1test2] đánh dấu dòng tiếp theo là kết quả bộ test thứ 2 của bài tập 1, <output1.2> là kết quả và có dạng như các ví dụ sau:

3	Kết quả của các thuật toán tìm kiếm. Trả về vị trí $p \in \{0, 1, \dots\}$ của phần tử cần tìm trong mảng. Trả về -1 nếu không tìm thấy.
0 0 1 3	Mảng sau khi sắp xếp tăng dần bằng thuật toán tương ứng.

### **Đề bài 1:**

Thuật toán linear search trên mảng.

### **Đề bài 2:**

Thuật toán binary search trên mảng.

### **Đề bài 3:**

Thuật toán interchange sort trên mảng.

### **Đề bài 4:**

Thuật toán bubble sort trên mảng.

### **Đề bài 5:**

Thuật toán shaker sort trên mảng.

**Đề bài 6:**

Thuật toán insertion sort trên mảng.

**Đề bài 7:**

Thuật toán shell sort trên mảng.

**Đề bài 8:**

Thuật toán selection sort trên mảng.

**Đề bài 9:**

Thuật toán heap sort trên mảng.

**Đề bài 10:**

Thuật toán quick sort trên mảng.

**Đề bài 11:**

Thuật toán merge sort trên mảng.

**Đề bài 12:**

Thuật toán radix sort trên mảng.

*Ghi chú:* cấu trúc dữ liệu mảng có thể là mảng tĩnh, động hoặc do sinh viên tự định nghĩa. Các file mã nguồn cần ghi thông tin sinh viên, mô tả hàm và chú thích cho các dòng code không đơn giản (xem lại file CTDL2015.PDF trên nhóm thảo luận). Sinh viên sẽ bị trừ điểm nếu bài làm không trình bày theo quy định.

Chúc các bạn làm tốt.