Bài tập thực hành 2: Search + Sort

- GVHDTH: Nguyễn Bảo Long
- Email: baolongnguyen.mac@gmail.com

1. Mục tiêu & Đánh giá

- Đối với bài thực hành này, sinh viên thực hiện tìm hiểu, cài đặt và viết báo cáo về các thuật toán search và sort (**theo thứ tự tăng dần**) đã học.
- Tìm hiểu và trình bày được các thuật toán - 6đ.
- Cài đặt được các thuật toán đã tìm hiểu - 4đ.
- Các bạn có thể tham khảo playlist này của thầy Nguyễn Tấn Trần Minh Khang.

2. Yêu cầu

- Sinh viên lập nhóm 2 người.
- Sinh viên thực hiện các yêu cầu đối với 2 nhóm thuật toán:
  - Search: Linear search, Binary search.
  - Sort: **Selection Sort**, **Insertion Sort**, Binary Insertion Sort, **Bubble Sort**, Shaker Sort, Shell Sort, **Heap Sort**, **Merge Sort**, **Quick Sort**, Counting Sort, Radix Sort, Flash Sort. Các thuật toán in đậm là các thuật toán bắt buộc cài đặt được.

2.1. Tìm hiểu & Trình bày thuật toán

- Trình bày ý tưởng chung.
- Trình bày mã giả.
- Đưa ra nhận xét về thuật toán: Các trường hợp tốt nhất, trung bình, tệ nhất sẽ có độ phức tạp như thế nào? Đưa ra chứng minh đối với trường hợp phức tạp nhất.
- Trình bày một ví dụ đơn giản (sort/search một mảng khoảng 5 phần tử) để minh họa cho mã giả. Lưu ý, các thuật toán nên lấy chung một mảng (ngoại trừ binary search, vì phải chuẩn bị một mảng tăng dần).

2.2. Cài đặt

- Cài đặt các thuật toán search.
- Cài đặt các thuật toán sort.
- Đối với các thuật toán sort, các bạn thí nghiệm với 3 tình trạng dữ liệu: đã sắp xếp tăng dần (sorted), có thứ tự ngược (mảng giảm dần - reversed), ngẫu nhiên (random). Với mỗi tình trạng,

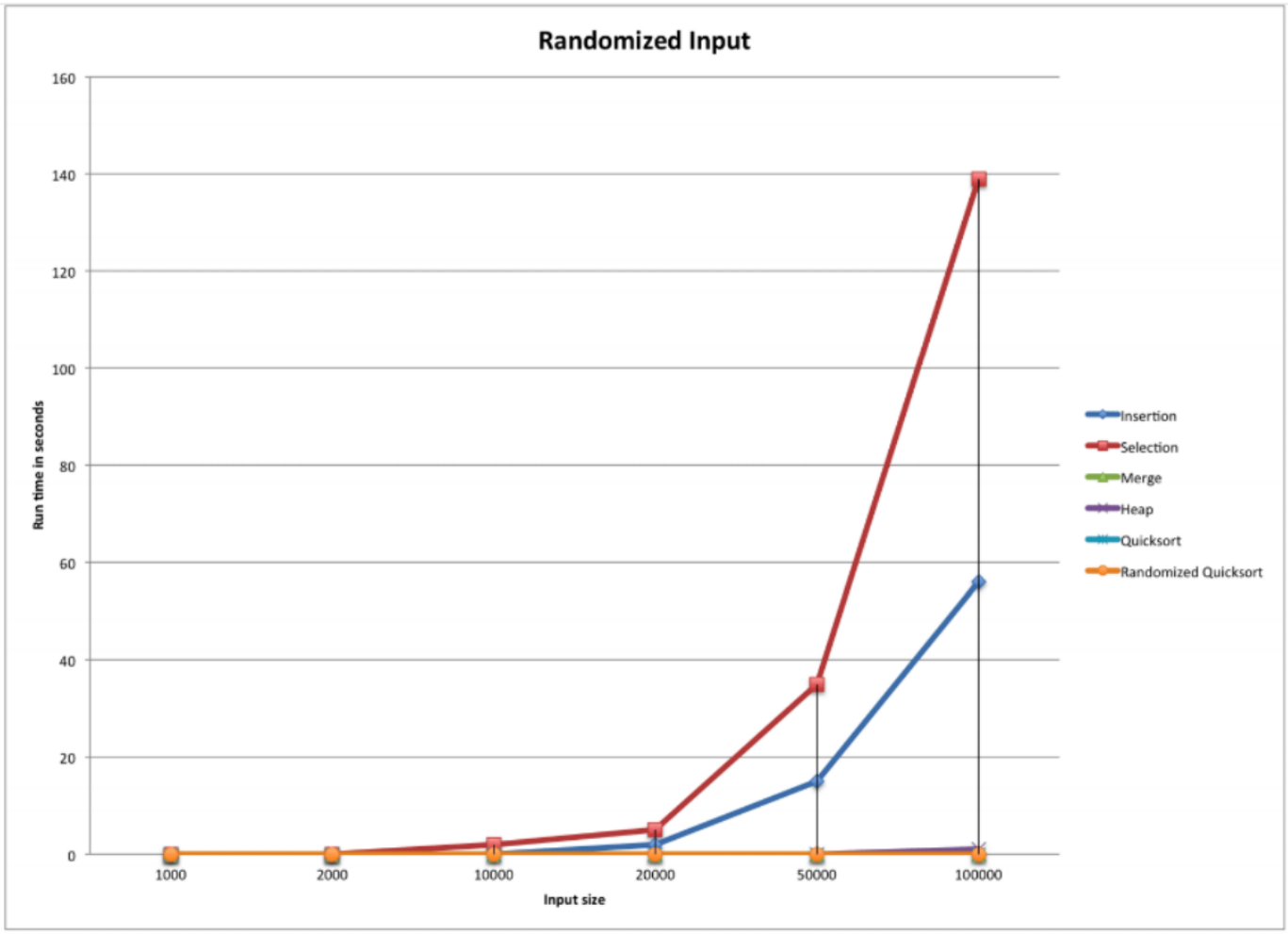
các bạn khảo sát 5 kích thước dữ liệu: 1000, 3000, 10.000, 30.000, 100.000.

```
for state in state_set:
    for size in size_set:
        for sort_algo in algorithm_set:
            # tạo mảng có tình trạng state và kích thước size
            array = create_array(state, size)

            # đo thời gian chạy của thuật toán
            timer.start()
            # sắp xếp tăng dần
            result = sort_algo(array)
            timer.stop()

            # lưu kết quả sort xuống file
            write_file(result, filename='<sort_algo>_<state>_<size>.txt')

            # lưu kết quả thời gian xuống file
            write_time(timer, filename='Runtime.csv')
```

- File `<sort_algo>_<state>_<size>.txt` ghi nhận kết quả sau khi sắp xếp. Các phần tử trong mảng được đặt cách nhau bởi 1 khoảng trắng.
- File `Runtime.csv` ghi nhận thời gian chạy của từng thuật toán. File này có dạng như sau:

	A	B	C	D	E	F	G	H
1	Input State	Input Size	Bubble	Selection	Insertion	Quick	Merge	Heap
2	Random	1000	0,1	0,1	0,1	0,1	0,1	0,1
3	Random	3000	0,2	0,2	0,2	0,2	0,2	0,2
4	Random	10000	0,3	0,3	0,3	0,3	0,3	0,3
5	Random	30000	0,4	0,4	0,4	0,4	0,4	0,4
6	Sorted	1000	0,5	0,5	0,5	0,5	0,5	0,5
7	Sorted	3000	0,6	0,6	0,6	0,6	0,6	0,6
8	Sorted	10000	0,7	0,7	0,7	0,7	0,7	0,7
9	Sorted	30000	0,8	0,8	0,8	0,8	0,8	0,8

- Vẽ biểu đồ thể hiện thời gian chạy của từng tình trạng dữ liệu. Trục tung thể hiện thời gian chạy, trục hoành thể hiện kích thước bộ dữ liệu. Màu của đường biểu diễn thể hiện tên thuật toán. Như vậy, sẽ có 3 biểu đồ có dạng như sau:



- Nhận xét về độ nhanh chậm của các thí nghiệm. Giải thích tại sao.

3. Quy định nộp bài

- Thời gian làm bài: 3 tuần.
- Sinh viên nộp bài dưới dạng file nén `<MSSV1>_<MSSV2>.zip`. Bên trong gồm có:
  - `./Code`: Chứa source code.
  - `Report.pdf`: Chứa báo cáo. Báo cáo cần có trang bìa, mục lục, phân công công việc, tỷ lệ hoàn thành, tự đánh giá bài nộp trên thang 10đ và nội dung bài tập được yêu cầu.
  - `Runtime.csv`: Chứa kết quả đo thời gian.
  - `./Results`: Chứa các file `*.txt` chứa kết quả chạy.
- Khuyến khích viết báo cáo dưới dạng Latex.
- Trích nguồn không đầy đủ trong báo cáo.
- Không chép bài.
- Vi phạm các quy định trên $\rightarrow$ 0đ.