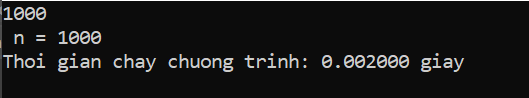
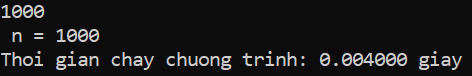
1. Giả sử sử dụng hàm Random trong C/C++ để phát sinh ngẫu nhiên dãy số nguyên có kích thước 100, 1.000, 5.000, 10.000, 50.000 số. Hãy cài đặt các giải thuật sắp xếp để sắp dãy số nguyên đã cho theo thứ tự tăng dần và nhận xét về thời gian thực hiện của các giải thuật.

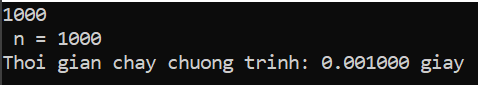
**- Kích thước 100:** Các thuật toán đều có thời gian chạy là 0.00000s

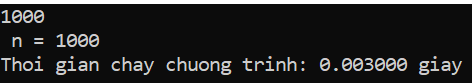
**- Kích thước 1000:**

**+ Selection Sort:**

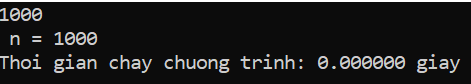
****

**+ Interchange Sort:**

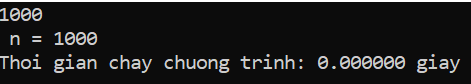
**** **+ Insertion Sort:**

****

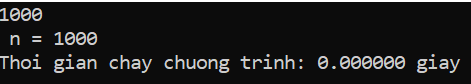
**+ Bubble Sort:**

****

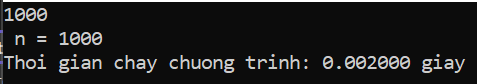
**+ Heap Sort:**

****

**+ Shell Sort:**

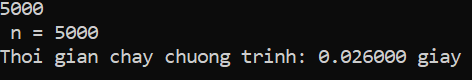
****

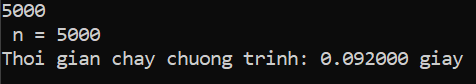
**+ Quick Sort:**

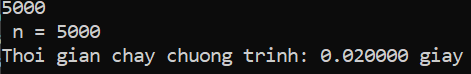
****

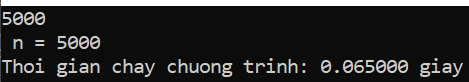
**+ Merge Sort:**

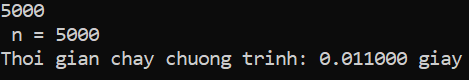
**- Kích thước 5000:**

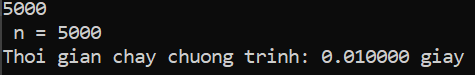
**+ Selection Sort:** 

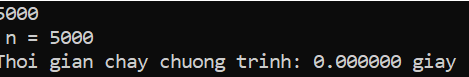
**+ Interchange Sort: **

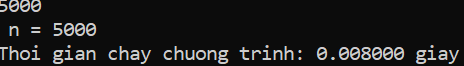
**+ Insertion Sort: **

**+ Bubble Sort: **

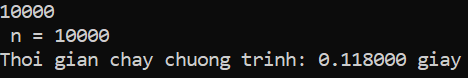
**+ Heap Sort: **

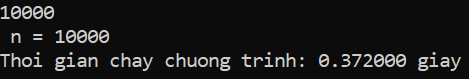
**+ Shell Sort: **

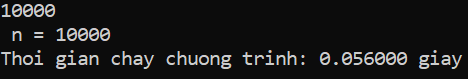
**+ Quick Sort: **

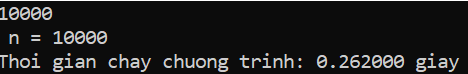
**+ Merge Sort: **

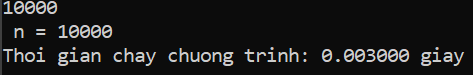
**- Kích thước 10000:**

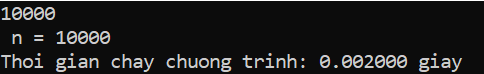
**+ Selection Sort:** 

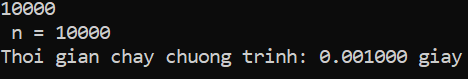
**+ Interchange Sort: **

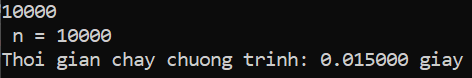
**+ Insertion Sort: **

**+ Bubble Sort: **

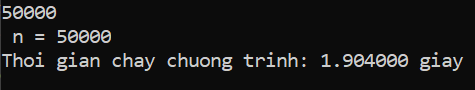
**+ Heap Sort: **

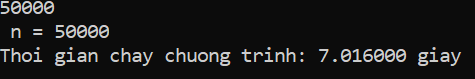
**+ Shell Sort: **

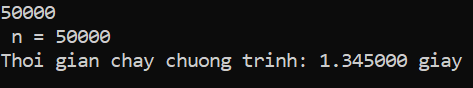
**+ Quick Sort: **

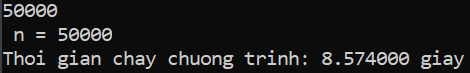
**+ Merge Sort:** ****

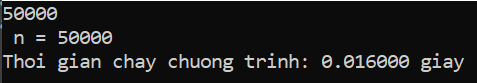
**- Kích thước 50000:**

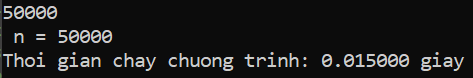
**+ Selection Sort:** 

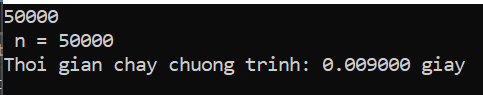
**+ Interchange Sort: **

**+ Insertion Sort: **

**+ Bubble Sort: **

**+ Heap Sort: **

**+ Shell Sort: **

**+ Quick Sort: **

- Bubble sort, Selection sort, Interchange Sort và Insertion sort đều có độ phức tạp là **O(n2)**, thời gian thực hiện của chúng sẽ tăng nhanh theo kích thước của dãy số. Khi sắp xếp các dãy số nguyên có kích thước lớn, thời gian thực hiện của các giải thuật này sẽ rất lâu và không hiệu quả.  
- Merge sort và Quick sort độ phức tạp là **O(n\*log n)**, thời gian thực hiện của chúng không tăng nhanh bằng các giải thuật có độ phức tạp O(n2). Khi sắp xếp các dãy số nguyên có kích thước lớn, thời gian thực hiện của các giải thuật này vẫn nhanh và hiệu quả hơn. Tuy nhiên, giải thuật sắp xếp trộn (Merge sort) sử dụng thêm bộ nhớ phụ để thực hiện quá trình trộn, do đó khi sắp xếp các dãy số nguyên có kích thước rất lớn, giải thuật này sẽ sử dụng rất nhiều bộ nhớ phụ và có thể gây ra sự cố về bộ nhớ.  
- Heap Sort: Giải thuật này sử dụng cấu trúc dữ liệu heap để sắp xếp tập dữ liệu. Độ phức tạp của nó là O(n log n), tuy nhiên thời gian thực hiện có thể bị ảnh hưởng bởi việc tạo heap ban đầu.  
- Shell Sort: Đây là một giải thuật sắp xếp có thể nhanh hơn Insertion Sort và Bubble Sort. Độ phức tạp của nó là O(n log n), nhưng tốc độ thực hiện phụ thuộc vào cách lựa chọn khoảng cách giữa các phần tử.