Lớp/Class: IT003.P21.CTTN

**BÁO CÁO KẾT QUẢ THỬ NGHIỆM**

**EXPERIMENT REPORT**

**Sinh viên thực hiện: Nguyễn Đức Khiêm**

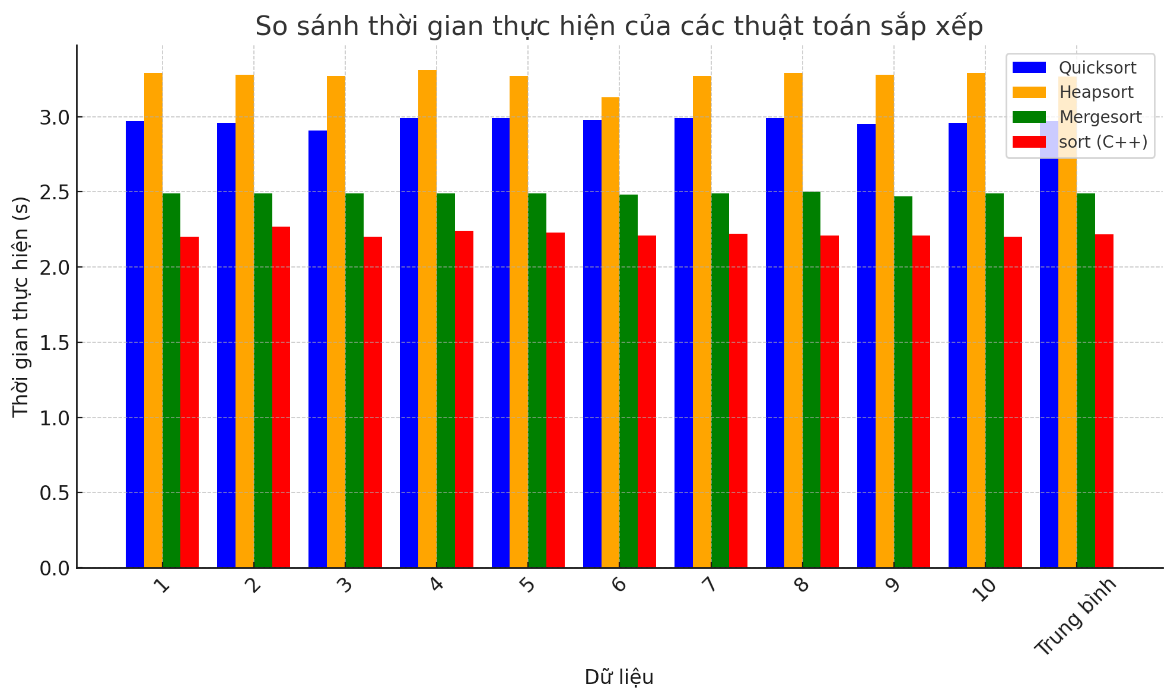
**Student: Nguyen Duc Khiem**

**Nội dung báo cáo: Thực nghiệm các thuật toán sắp xếp**

**Report Content: Experimenting with Sorting Algorithms**

1. ***Kết quả thử nghiệm/ Experimental Results***
   1. ***Bảng thời gian thực hiện[[1]](#footnote-1)/ Execution Time Table***
      * Lưu ý: “Dữ liệu 1” bao gồm 10 mảng với mỗi mảng gồm 1 000 000 phần tử. Mảng 1 đã tang dần. mảng 2 đã giảm dần.
      * Note: "Data 1" includes 10 arrays, each containing 1,000,000 elements. Array 1 is sorted in ascending order, while array 2 is sorted in descending order.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Dữ liệu/Data** | **Thời gian thực hiện / Execution time (s)** | | | |
| **Quicksort** | **Heapsort** | **Mergesort** | **sort (C++)** |
| 1 | 2.97 | 3.29 | 2.49 | 2.20 |
| 2 | 2.96 | 3.28 | 2.49 | 2.27 |
| 3 | 2.91 | 3.27 | 2.49 | 2.20 |
| 4 | 2.99 | 3.31 | 2.49 | 2.24 |
| 5 | 2.99 | 3.27 | 2.49 | 2.23 |
| 6 | 2.98 | 3.13 | 2.48 | 2.21 |
| 7 | 2.99 | 3.27 | 2.50 | 2.22 |
| 8 | 2.99 | 3.29 | 2.50 | 2.21 |
| 9 | 2.95 | 3.28 | 2.47 | 2.21 |
| 10 | 2.96 | 3.29 | 2.49 | 2.20 |
| Trung bình/ Average | 2.97 | 3.268 | 2.489 | 2.219 |

* 1. ***Biểu đồ (cột) thời gian thực hiện / Barchart***

1. ***Kết luận / Conclusion:***
   * + **Thuật toán sort() của C++ nhanh nhất** – Với thời gian trung bình khoảng **2.219 giây**, thuật toán này vượt trội hơn so với các thuật toán khác. Điều này cho thấy sự tối ưu hóa cao của thư viện C++ Standard Template Library (STL).
     + **Mergesort có hiệu suất ổn định** – Với thời gian trung bình **2.489 giây**, Mergesort có hiệu suất tương đối tốt và ít biến động giữa các lần chạy. Đây là một lựa chọn phù hợp cho các trường hợp cần độ ổn định cao.
     + **Quicksort hoạt động tốt nhưng không tối ưu nhất** – Với thời gian trung bình **2.97 giây**, Quicksort vẫn là một thuật toán mạnh mẽ, nhưng nó chậm hơn một chút so với Mergesort và sort() của C++. Tuy nhiên, Quicksort vẫn là lựa chọn phổ biến trong nhiều ứng dụng thực tế nhờ hiệu suất trung bình tốt và dễ triển khai.
     + **Heapsort chậm nhất** – Với thời gian trung bình **3.268 giây**, Heapsort có tốc độ chậm hơn đáng kể so với các thuật toán khác. Điều này là do chi phí thao tác trên heap cao hơn so với cách phân chia và chinh phục của Quicksort và Mergesort.
     + The sort() function in C++ is the fastest – With an average execution time of approximately 2.219 seconds, this algorithm outperforms the others. This demonstrates the high level of optimization in the C++ Standard Template Library (STL).
     + Mergesort has stable performance – With an average time of 2.489 seconds, Mergesort provides relatively good performance with minimal variations across runs. This makes it a suitable choice for cases requiring high stability.
     + Quicksort performs well but is not the most optimal – With an average execution time of 2.97 seconds, Quicksort remains a powerful algorithm but is slightly slower than Mergesort and the C++ sort(). However, Quicksort is still widely used in practice due to its good average performance and ease of implementation.
     + Heapsort is the slowest – With an average time of 3.268 seconds, Heapsort is significantly slower than the other algorithms. This is due to the higher cost of heap operations compared to the divide-and-conquer approach of Quicksort and Mergesort.

**Kết luận tổng quát/** **General Conclusion:**

* + - Nếu mục tiêu là tốc độ tối đa, sử dụng sort() của C++ là lựa chọn tốt nhất. Nếu cần một thuật toán ổn định, Mergesort là một lựa chọn phù hợp. Quicksort vẫn là một thuật toán hiệu quả nhưng có thể bị đánh bại bởi các phương pháp tối ưu hơn. Heapsort tuy có lợi ích về độ ổn định nhưng hiệu suất kém hơn, khiến nó ít được sử dụng trong thực tế.
    - If maximum speed is the goal, using C++ sort() is the best choice. If a stable algorithm is required, Mergesort is a suitable option. Quicksort remains an efficient algorithm but can be outperformed by more optimized methods. While Heapsort has the advantage of stability, its inferior performance makes it less commonly used in practice.

1. Số liệu chỉ mang tính minh họa [↑](#footnote-ref-1)