

Thực nghiệm thuật toán Tim Sort

1. Mục tiêu thực nghiệm

- Kiểm tra **correctness** của thuật toán Tim Sort.
- Đánh giá **adaptiveness** của thuật toán đối với các dạng dữ liệu đầu vào khác nhau.
- So sánh **độ phức tạp thời gian thực nghiệm** với **độ phức tạp lý thuyết**.

2. Các bài kiểm tra thực nghiệm

2.1. Correctness Test

Mục tiêu của bài kiểm tra này là xác minh thuật toán Tim Sort có cho ra kết quả sắp xếp đúng trong các trường hợp thông thường và biên hay không.

Thuật toán được kiểm tra với các trường hợp sau: Mảng rỗng, mảng chỉ chứa một phần tử, mảng gồm các phần tử có giá trị bằng nhau, mảng đã được sắp xếp tăng dần, mảng đã được sắp xếp giảm dần, mảng với các phần tử ngẫu nhiên.

Kết quả:

Thuật toán Tim Sort cho kết quả đúng trong tất cả các trường hợp kiểm tra.

2.2. Adaptiveness Test

Mục tiêu của bài kiểm tra này là đánh giá khả năng thích nghi của Tim Sort đối với các dạng dữ liệu đầu vào khác nhau nhưng có **cùng kích thước mảng**.

Các trường hợp được xét gồm:

- **Best case:** Mảng đầu vào đã được sắp xếp tăng dần hoàn toàn. Thuật toán nhanh chóng phát hiện một run tăng dần có độ dài bằng kích thước mảng.
- **Worst case:** Mảng gồm số lượng run đạt mức tối đa; trong trường hợp này, mảng được cấu tạo từ các run có độ dài rất nhỏ (ví dụ: các run độ dài 2).
- **Average case:** Các phần tử trong mảng được sinh ngẫu nhiên.

Kết quả đo thời gian (đơn vị: ms)

===== RESULT TABLE =====			
Best Case	Avg Case	Worst Case	

0.375	23.882	8.989	

Nhận xét

- **Best case** cho thời gian chạy nhỏ nhất do thuật toán tận dụng tối đa các run tăng dần đã tồn tại sẵn trong dữ liệu đầu vào.
- **Average case** cho thời gian chạy lớn hơn **worst case** mặc dù cả hai đều có độ phức tạp lý thuyết là $O(n \log n)$. Nguyên nhân:
 - Trong **worst case**, cấu trúc dữ liệu đầu vào có tính quy luật cao, giúp hệ thống tối ưu quá trình thực thi.
 - Trong **average case**, dữ liệu mang tính ngẫu nhiên, làm giảm hiệu quả của các cơ chế tối ưu ở mức hệ thống.

2.3. Performance Test

Mục tiêu của bài kiểm tra này là xác minh xem độ phức tạp thời gian của Tim Sort trong thực nghiệm có phù hợp với phân tích lý thuyết hay không.

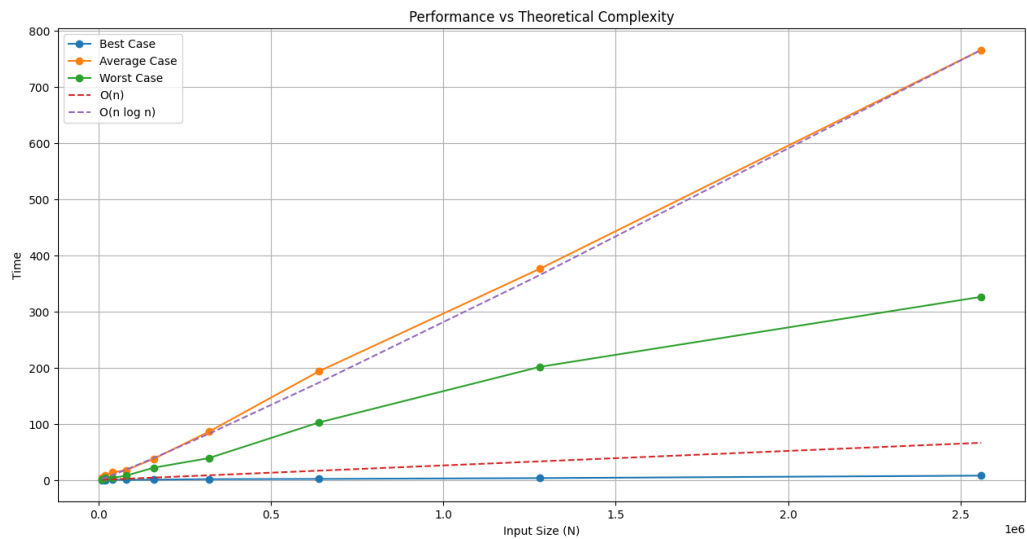
Thuật toán được chạy trên các tập dữ liệu **best case, average case, worst case** với kích thước mảng tăng dần n_1, n_2, n_3, \dots , từ đó ghi nhận các thời gian chạy tương ứng t_1, t_2, t_3, \dots và so sánh xu hướng tăng trưởng.

Kết quả đo thời gian (đơn vị: ms)

===== RESULT TABLE =====			
Size (N)	Best Case	Avg Case	Worst Case

10000	0.033	3.435	1.395
20000	0.130	8.513	3.144
40000	0.314	13.288	3.607
80000	0.207	17.582	7.617
160000	0.494	37.611	21.796
320000	1.428	85.902	38.902
640000	1.809	193.718	102.500
1280000	3.243	376.154	201.471
2560000	7.680	766.014	326.179

3. Phân tích và nhận xét



Dựa trên biểu đồ so sánh với các hàm lý thuyết:

- Đường **Best case** có xu hướng tăng gần tương đương với hàm $O(n)$.
- Đường **Average case** bám sát hàm $O(n \log n)$, phù hợp với phân tích lý thuyết của thuật toán.
- Đường **Worst case** nằm giữa $O(n)$ và $O(n \log n)$ trong biểu đồ thực nghiệm. Hiện tượng này có thể được giải thích bởi các yếu tố tối ưu ở mức hệ thống khiến thời gian thực thi thực tế thấp hơn so với kịch bản lý thuyết thuần túy.