*CÂU HỎI LÝ THUYẾT CHƯƠNG 3*

***Họ và tên****: Nguyễn Thị Thanh Hằng*

***MSSV****: 1654050032*

***Nhóm****: 02*

**Bài làm**:

***Câu 1****: Trong các phương pháp sắp xếp thứ tự đã học , phương pháp nào tối ưu nhất, kém tối ưu nhất? Tại sao?*

Trong các pp sắp xếp thứ tự đã học , phương pháp tối ưu nhất là phương pháp Quick Sort, phương pháp kém tối ưu nhất là Bubble Sort.

Lý do:

* Quick Sort có độ phức tạp thuộc lớp O(nlogn) là một trong những thuật toán sắp xếp nhanh nhất trong số các giải thuật sắp xếp cơ bản, cùng với việc sử dụng không gian O (log n) khiêm tốn của nó, Quicksort là một trong những thuật toán sắp xếp phổ biến nhất và có sẵn trong nhiều thư viện lập trình tiêu chuẩn.
* Bubble Sort có độ phức tạp trung bình là O(n^2) được đánh giá là chưa tốt, thời gian chạy rất lâu trong số các giải thuật sắp xếp cơ bản .

***Câu 2****: Trong các phương pháp tìm kiếm đã học, trường hợp nào thì cả 2 phương pháp đều như nhau ? Giải thích tại sao?*

Để cả 2 phương pháp tìm kiếm tuần tự và tìm kiếm nhị phân đều như nhau ,theo em trong trường hợp mảng tìm kiếm chỉ có một phần tử duy nhất.

Lý do:

Trong trường hợp dãy chỉ có 1 phần tử thì cả 2 phương pháp tìm kiếm chỉ thực hiện 1 phép so sánh duy nhất hoặc là tìm thấy phần tử cần tìm hoặc là không.

***Câu 3****: Ngoài các phương pháp xếp thứ tự đã học,hãy tìm hiểu thêm một phương pháp xếp thứ tự khác,giới thiệu sơ và giải thích?*

* **Phương pháp Merger Sort** ( phương pháp sắp xếp trộn ):
* **Ý tưởng** :chúng ta sẽ chia mảng lớn thành những mảng con nhỏ hơn bằng cách chia đôi mảng lớn và chúng ta tiếp tục chia đôi các mảng con cho tới khi mảng con nhỏ nhất chỉ còn 1 phần tử. Sau đó chúng ta sẽ tiếng hành so sánh 2 mảng con có cùng mảng cơ sở (khi chúng ta chia đôi mảng lớn thành 2 mảng con thì mảng lớn đó chúng ta gọi là mảng cơ sở của 2 mảng con đó) khi so sánh chúng sẽ vừa sắp xếp vừa ghép 2 mảng con đó lại thành mảng cơ sở, chúng ta tiếp tục so sánh và ghép các mảng con lại đến khi còn lại mảng duy nhất thì đó là mảng đã được sắp xếp.

Giả sử có hai danh sách đã được sắp xếp a[1..m] và b[1..n]. Ta có thể trộn chúng lại thành một danh sách mới c[1..m+n] được sắp xếp theo cách sau:

* So sánh hai phần tử đứng đầu của hai danh sách, lấy phần tử nhỏ hơn cho vào danh sách mới. Tiếp tục như vậy cho tới khi một trong hai danh sách là rỗng.
* Khi một trong hai danh sách là rỗng ta lấy phần còn lại của danh sách kia cho vào cuối danh sách mới.