**CHƯƠNG 3**

*Câu hỏi:*

1. Trong các phương pháp sắp xếp thứ tự đã học:

- Phương pháp sắp xếp tối ưu nhất là *Buble Sort* bởi theo thống kê cho ra thì tốc độ nhanh nhất là O(n).

-Kém tối ưu nhất là *Selection Sort*, Selction Sort cho ra tốc dộ chậm hơn trong đa số trường hợp với độ phức tạp là O(n2). Do đó Selection Sort chỉ nên dùng để sắp xếp cho số lượng phần tử không quá nhiều.

1. Khi cả 2 phương pháp tìm kiếm là như nhau trong trường hợp tìm kiếm tốt nhất, là khi độ phức tạp bằng 0.

=> Vì với trường hợp tốt nhất, chỉ có 1 lần xét trong cả 2 phương pháp, tìm kiếm tuần tự là ở đầu và tìm kiếm nhị phân là ở giữa.

1. *Radix Sort* là một thuật toán sắp xếp không so sánh dựa trên nguyên tắc phân loại thư của bưu điện.

\*Thuật toán này được thực hiện dựa trên ý tưởng nếu một danh sách đã được sắp xếp hoàn chỉnh thì từng phần tử cũng sẽ được sắp xếp hoàn chỉnh dựa trên giá trị của các phần tử đó. Thuật toán này yêu cầu danh sách cần được sắp xếp để có thể so sánh thứ tự các vị trí vì thế sắp xếp theo cơ số không giới hạn ở tập số nguyên (ta có thể dễ dàng đưa dạng xâu về cơ số nhị phân).

*Phân loại*: Thuật toán sắp xếp

*Cấu trúc dữ liệu*: Array

*Phức tạp thời gian:* O(wn)

*Phức tạp không gian*: O(w+n)

\*Các bước thực hiện thuật toán như sau:

Bước 1 : k cho biết chữ số dùng để phân loại hiện hành

k = 0; (k = 0: hàng đơn vị; k = 1: hàng chục)

Bước 2 : Tạo các lô chứa các loại phần tử khác nhau

Khởi tạo 10 lô B0, B1, .. , B9 rỗng;

Bước 3 :

For i = 1 .. n do

Ðặt ai vào lô Bt với t = chữ số thứ k của ai;

Bước 4 :

Nối B0, B1, .. , B9 lại (theo đúng trình tự) thành a.

*Bước 5* :

k = k+1;

Nếu k < m thì trở lại bước 2.

Ngược lại: Dừng