



ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

CHƯƠNG IV

TÌM KIẾM VÀ SẮP XẾP



NỘI DUNG CHƯƠNG IV

- I. NHU CẦU TÌM KIẾM, SẮP XẾP**
- II. CÁC GIẢI THUẬT TÌM KIẾM**
- III. CÁC GIẢI THUẬT SẮP XẾP**
- IV. CẤU TRÚC HÀNG ĐỢI ƯU TIÊN**



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHÁT BIỂU BÀI TOÁN

Cho danh sách **A** gồm n phần tử **a_0, a_1, \dots, a_{n-1}**

Hoán đổi vị trí của các phần tử a_i và a_j sao cho đảm bảo thứ tự **\mathcal{R}** trong **A**, nghĩa là

$$a_i \mathcal{R} a_j, \forall i < j$$



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHÂN LOẠI SẮP XẾP

Các giải thuật sắp xếp có thể phân loại theo nhiều tiêu chí:

a) Tính chất của danh sách A:

- Toàn bộ phần tử của A được xử lý đồng thời trong quá trình sắp xếp → **Offline Sorting**
 - Selection Sort
 - Bubble Sort
 - Quick Sort
 -



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHÂN LOẠI SẮP XẾP

Các giải thuật sắp xếp có thể phân loại theo nhiều tiêu chí:

a) Tính chất của danh sách A:

- Từng phần tử của A được sắp xếp tuần tự mà không cần biết trước toàn bộ A → **Online**

Sorting

- Insertion Sort
- Tree Sort (tạo Binary Search Tree)



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHÂN LOẠI SẮP XẾP

Các giải thuật sắp xếp có thể phân loại theo nhiều tiêu chí:

b) Trật tự của kết quả sắp xếp:

- Thứ tự trước/sau của các phần tử có cùng giá trị khóa không đổi so với ban đầu → **Stable Sorting**. Ví dụ:

- Bubble Sort:

- Trước khi sắp xếp: $A = \{1, 5_1, 2, 5_2, 3, 5_3\}$
- Sau khi sắp xếp (tăng dần): $A = \{1, 2, 3, 5_1, 5_2, 5_3\}$



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHÂN LOẠI SẮP XẾP

Các giải thuật sắp xếp có thể phân loại theo nhiều tiêu chí:

b) Trật tự của kết quả sắp xếp:

- Thứ tự trước/sau của các phần tử có cùng giá trị khóa thay đổi so với ban đầu → **Unstable Sorting**. Ví dụ:

- Interchange Sort:

- Trước khi sắp xếp: $A = \{1, 5_1, 2, 5_2, 3, 5_3\}$
- Sau khi sắp xếp (tăng dần): $A = \{1, 2, 3, 5_2, 5_1, 5_3\}$



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHÂN LOẠI SẮP XẾP

Các giải thuật sắp xếp có thể phân loại theo nhiều tiêu chí:

c) Nơi lưu trữ chính của danh sách:

➤ Toàn bộ danh sách A được lưu trữ trên RAM trong quá trình sắp xếp → **Internal Sorting.**

- Interchange Sort
- Insertion Sort
- Quick Sort
- ...



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHÂN LOẠI SẮP XẾP

Các giải thuật sắp xếp có thể phân loại theo nhiều tiêu chí:

c) Nơi lưu trữ chính của danh sách:

- Toàn bộ danh sách A được lưu trữ trên bộ nhớ ngoài (HDD) trong quá trình sắp xếp do kích thước danh sách quá lớn → **External Sorting.**
 - Merge Sort



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHỌN TRỰC TIẾP

Từ khóa: Selection Sort

Phân tích: Giả sử danh sách $A = \{a_0, a_1, \dots, a_{n-1}\}$ đã có thứ tự \mathcal{R} . Khi đó:

- a_0 là phần tử nhỏ nhất theo \mathcal{R} trong A
- a_1 là phần tử nhỏ nhất theo \mathcal{R} trong $A \setminus \{a_0\}$
- a_2 là phần tử nhỏ nhất theo \mathcal{R} trong $A \setminus \{a_0, a_1\}$
- ...



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHỌN TRỰC TIẾP

Ý tưởng: Chọn phần tử nhỏ thứ i theo thứ tự \mathcal{R} trong danh sách A và đặt vào vị trí i của danh sách.



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHỌN TRỰC TIẾP

Thuật toán:

Đầu vào: $A = \{a_0, a_1, \dots, a_{n-1}\}$ chưa có thứ tự \mathfrak{R}

Đầu ra: $A = \{a_0, a_1, \dots, a_{n-1}\}$ đã có thứ tự \mathfrak{R}



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHỌN TRỰC TIẾP

Thuật toán:

```
i ← 0
while i < n - 1
    min ← i, j ← i+1
    while j < n
        if A[j] < A[min] then min ← j
        j ← j+1
    swap(A[i], A[min])
    i ← i + 1
```



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHỌN TRỰC TIẾP

Quá trình tính toán:

Giả sử $A=\{3,2,5,1,4\}$ và thứ tự cần sắp xếp < (tăng dần)

i=0	j=1	2	3	4
3	2	5	1	4

min=0



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHỌN TRỰC TIẾP

Quá trình tính toán:

Giả sử $A=\{3,2,5,1,4\}$ và thứ tự cần sắp xếp < (tăng dần)

i=0	1	j=2	3	4
3	2	5	1	4

min=1



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHỌN TRỰC TIẾP

Quá trình tính toán:

Giả sử $A=\{3,2,5,1,4\}$ và thứ tự cần sắp xếp < (tăng dần)

i=0	1	2	j=3	4
3	2	5	1	4

min=1



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHỌN TRỰC TIẾP

Quá trình tính toán:

Giả sử $A=\{3,2,5,1,4\}$ và thứ tự cần sắp xếp < (tăng dần)

i=0	1	2	3	j=4
3	2	5	1	4

min=3



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHỌN TRỰC TIẾP

Quá trình tính toán:

Giả sử $A=\{3,2,5,1,4\}$ và thứ tự cần sắp xếp < (tăng dần)

$i=0$	1	2	3	$j=4$
3	2	5	1	4

$\text{min}=3$

Hoán đổi giá trị
 $A[i]$ và $A[\text{min}]$



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHỌN TRỰC TIẾP

Quá trình tính toán:

Giả sử $A=\{3,2,5,1,4\}$ và thứ tự cần sắp xếp < (tăng dần)

0	i=1	j=2	3	4
1	2	5	3	4

min=1



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHỌN TRỰC TIẾP

Quá trình tính toán:

Giả sử $A=\{3,2,5,1,4\}$ và thứ tự cần sắp xếp < (tăng dần)

0	i=1	2	j=3	4
1	2	5	3	4

min=1



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHỌN TRỰC TIẾP

Quá trình tính toán:

Giả sử $A=\{3,2,5,1,4\}$ và thứ tự cần sắp xếp < (tăng dần)

0	i=1	2	3	j=4
1	2	5	3	4

min=1



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHỌN TRỰC TIẾP

Quá trình tính toán:

Giả sử $A=\{3,2,5,1,4\}$ và thứ tự cần sắp xếp < (tăng dần)

0	$i=1$	2	3	4
1	2	5	3	4

$\text{min}=1$

Hoán đổi giá trị
 $A[i]$ và $A[\text{min}]$



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHỌN TRỰC TIẾP

Quá trình tính toán:

Giả sử $A=\{3,2,5,1,4\}$ và thứ tự cần sắp xếp < (tăng dần)

0	1	$i=2$	$j=3$	4
1	2	5	3	4

$\text{min}=2$



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHỌN TRỰC TIẾP

Quá trình tính toán:

Giả sử $A=\{3,2,5,1,4\}$ và thứ tự cần sắp xếp < (tăng dần)

0	1	$i=2$	3	$j=4$
1	2	5	3	4

$\text{min}=3$



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHỌN TRỰC TIẾP

Quá trình tính toán:

Giả sử $A=\{3,2,5,1,4\}$ và thứ tự cần sắp xếp < (tăng dần)

0	1	$i=2$	3	4
1	2	5	3	4

$\text{min}=3$

Hoán đổi giá trị
 $A[i]$ và $A[\text{min}]$



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHỌN TRỰC TIẾP

Quá trình tính toán:

Giả sử $A=\{3,2,5,1,4\}$ và thứ tự cần sắp xếp < (tăng dần)

0	1	2	$i=3$	$j=4$
1	2	3	5	4

$\text{min}=3$



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHỌN TRỰC TIẾP

Quá trình tính toán:

Giả sử $A=\{3,2,5,1,4\}$ và thứ tự cần sắp xếp < (tăng dần)

0	1	2	$i=3$	4
1	2	3	5	4

$\text{min}=3$

Hoán đổi giá trị
 $A[i]$ và $A[\text{min}]$



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHỌN TRỰC TIẾP

Quá trình tính toán:

Giả sử $A = \{3, 2, 5, 1, 4\}$ và thứ tự cần sắp xếp < (tăng dần)

0	1	2	3	4
1	2	3	4	5

Kết thúc



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHỌN TRỰC TIẾP

Cài đặt: (trên mảng) giả sử thứ tự là $<$ (tăng dần)

```
void selectionSort(int A[], int n) {  
    int min;  
    for (int i = 0; i < n-1; i++) {  
        min = i;  
        for (int j = i+1; j < n; j++)  
            if (A[j] < A[min]) min = j;  
        swap(A[i], A[min]);  
    }  
}
```

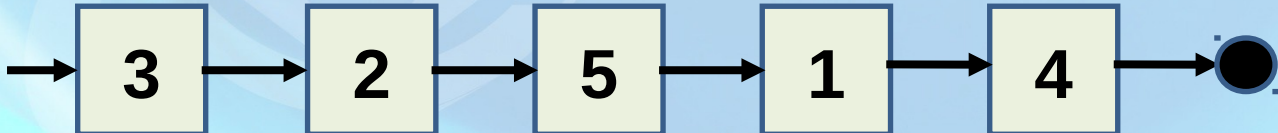


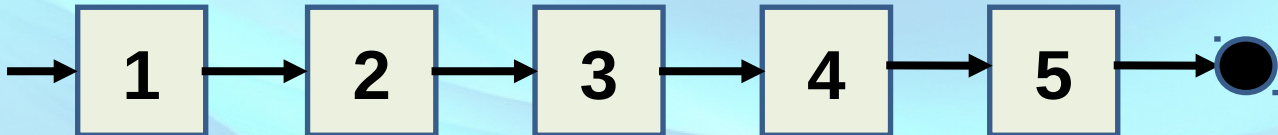

III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHỌN TRỰC TIẾP

Cài đặt: (trên danh sách liên kết đơn) giả sử thứ tự là $<$ (tăng dần)

Trường hợp 1: sắp xếp bằng cách thay đổi giá trị tại mỗi node

Chưa sắp xếp: 

Đã sắp xếp: 



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHỌN TRỰC TIẾP

```
void selectionSort(List A) {  
    Node *min, *i, *j;  
    i = A.pHead;  
    while (i) {  
        min = i; j = i->pNext;  
        while (j) {  
            if (j->info < min->info) min = j;  
            j = j->pNext;  
        }  
    }
```



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHỌN TRỰC TIẾP

```
swap(i->info, min->info);
```

```
i = i->pNext;
```

```
}
```

```
}
```




III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHỌN TRỰC TIẾP

Cài đặt: (trên danh sách liên kết đơn) giả sử thứ tự là $<$ (tăng dần)

Trường hợp 2: sắp xếp bằng cách thay đổi liên kết tại mỗi node

Chưa sắp xếp: →

Đã sắp xếp:



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHỌN TRỰC TIẾP

```
void selectionSort(List &A) {  
    Node *qmin, *i, *j, *h;  
    h = NULL; i = A.pHead;  
    while (i->pNext) {  
        qmin = h; j = i;  
        while (j->pNext) {  
            if (j->pNext->info < i->info) {  
                qmin = j; i = qmin->pNext;  
            }  
            j = j->pNext;  
        }  
    }  
}
```



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHỌN TRỰC TIẾP

```
addAfter(A, removeAfter(A, qmin), h);  
if (!h) h = A.pHead;  
else h = h->pNext;  
i = h->pNext;  
}  
}
```




III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHỌN TRỰC TIẾP

Đánh giá:

	TỐT NHẤT (đúng thứ tự)	TRUNG BÌNH (chưa có thứ tự)	XẤU NHẤT (thứ tự ngược)
Theo phép so sánh	$O(n^2)$	$O(n^2)$	$O(n^2)$
Theo phép gán giá trị khóa	$O(n)$	$O(n)$	$O(n)$



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHÈN TRỰC TIẾP

Từ khóa: Insertion Sort

Phân tích: Giả sử danh sách $A = \{a_0, a_1, \dots, a_{n-1}\}$ đã có thứ tự \mathcal{R} . Khi đó, để tạo danh sách A có n+1 phần tử có thứ tự \mathcal{R} , cần tìm vị trí k để chèn phần tử a_n sao cho đảm bảo

$$a_i \mathcal{R} a_n \quad \forall i < k$$



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHÈN TRỰC TIẾP

Ý tưởng:

- Danh sách chỉ có một phần tử luôn có thứ tự. Như vậy, $A_0 = \{a_0\}$ là danh sách có thứ tự \mathcal{R}
- Để sắp xếp $A = \{a_0, a_1, \dots, a_{n-1}\}$ theo thứ tự \mathcal{R} . Lần lượt lấy a_i ($i > 0$) trong A và thực hiện
 1. Bắt đầu từ cuối danh sách $A_{i-1} = \{a_0, \dots, a_{i-1}\}$, Tìm vị trí k đầu tiên thỏa điều kiện $a_k \mathcal{R} a_i$.
 2. Đẩy tất cả phần tử (nếu có) từ ngay sau vị trí k về bên phải 1 vị trí.
 3. Đưa vào a_i vị trí $k+1$ của A_{i-1} , A_{i-1} thành A_i



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHÈN TRỰC TIẾP

Thuật toán:

Đầu vào: $A = \{a_0, a_1, \dots, a_{n-1}\}$ chưa có thứ tự \mathfrak{R}

Đầu ra: $A = \{a_0, a_1, \dots, a_{n-1}\}$ đã có thứ tự \mathfrak{R}



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHÈN TRỰC TIẾP

Thuật toán:

```
i ← 1
while i < n
    e ← A[i]
    k ← i-1
    while (k > 0) and not (A[k] ⋈ e)
        A[k+1] ← A[k]
        k ← k-1
    A[k+1] ← e,
    i ← i+1
```



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHÈN TRỰC TIẾP

Quá trình tính toán:

Giả sử $A=\{3,2,5,1,4\}$ và thứ tự cần sắp xếp < (tăng dần).

	0	i=1	2	3	4
e=2	3	2	5	1	4
k=0					



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHÈN TRỰC TIẾP

Quá trình tính toán:

Giả sử $A=\{3,2,5,1,4\}$ và thứ tự cần sắp xếp < (tăng dần)

	0	i=1	2	3	4
e=2	3	3	5	1	4
k=-1					

Đưa phần tử e
vào vị trí k+1

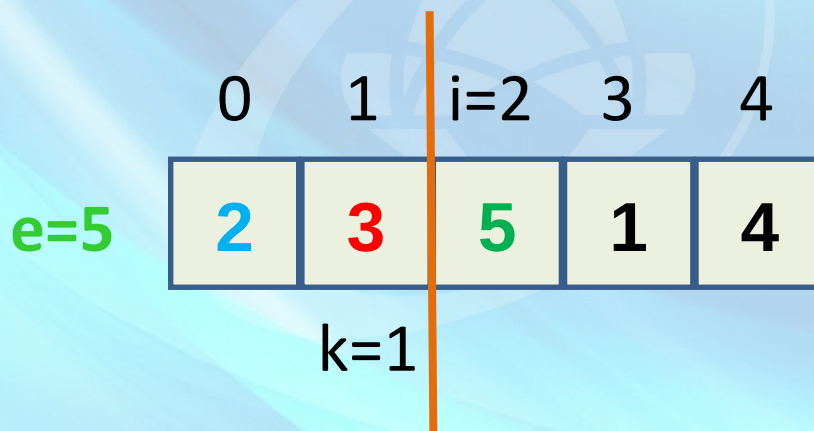


III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHÈN TRỰC TIẾP

Quá trình tính toán:

Giả sử $A=\{3,2,5,1,4\}$ và thứ tự cần sắp xếp $<$
(tăng dần)



Đưa phần tử e
vào vị trí k+1

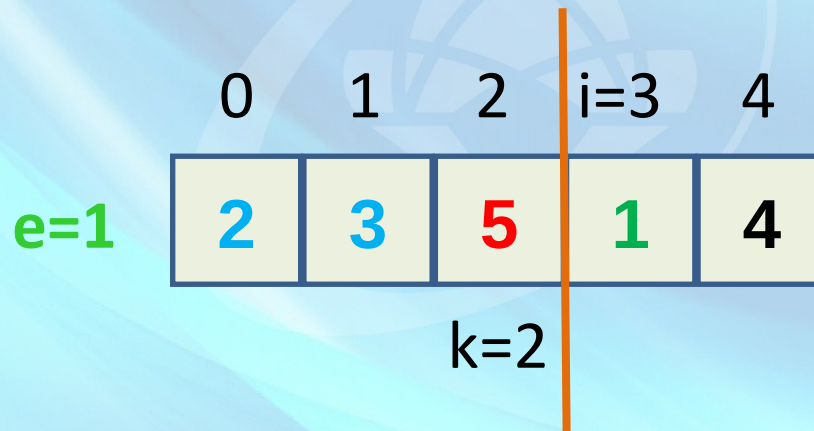


III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHÈN TRỰC TIẾP

Quá trình tính toán:

Giả sử $A=\{3,2,5,1,4\}$ và thứ tự cần sắp xếp < (tăng dần)



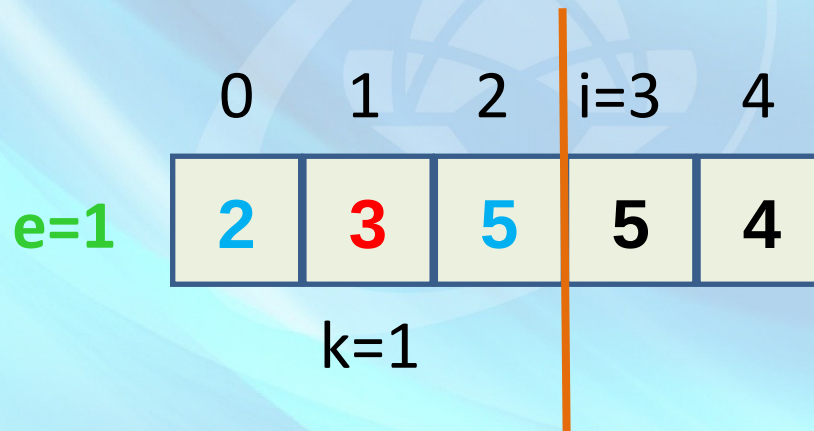


III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHÈN TRỰC TIẾP

Quá trình tính toán:

Giả sử $A=\{3,2,5,1,4\}$ và thứ tự cần sắp xếp < (tăng dần)



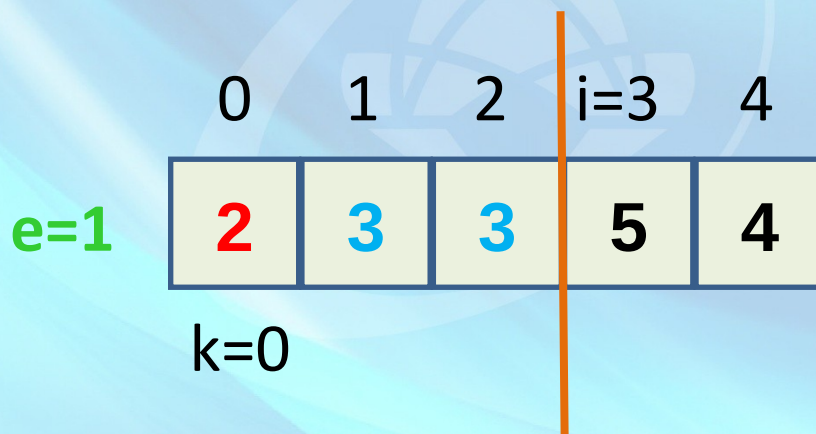


III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHÈN TRỰC TIẾP

Quá trình tính toán:

Giả sử $A=\{3,2,5,1,4\}$ và thứ tự cần sắp xếp < (tăng dần)





III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHÈN TRỰC TIẾP

Quá trình tính toán:

Giả sử $A=\{3,2,5,1,4\}$ và thứ tự cần sắp xếp < (tăng dần)

	0	1	2	$i=3$	4
$e=1$	2	2	3	5	4
$k=-1$					

Đưa phần tử e
vào vị trí $k+1$



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHÈN TRỰC TIẾP

Quá trình tính toán:

Giả sử $A=\{3,2,5,1,4\}$ và thứ tự cần sắp xếp < (tăng dần)



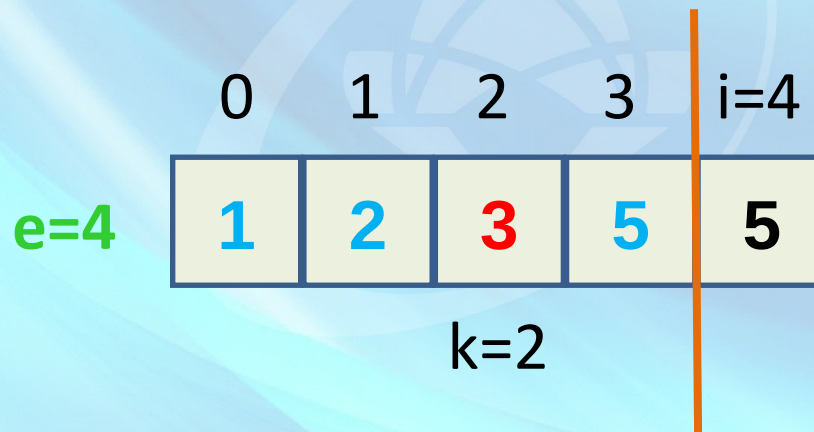


III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHÈN TRỰC TIẾP

Quá trình tính toán:

Giả sử $A=\{3,2,5,1,4\}$ và thứ tự cần sắp xếp < (tăng dần)



Đưa phần tử e
vào vị trí $k+1$



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHÈN TRỰC TIẾP

Quá trình tính toán:

Giả sử $A=\{3,2,5,1,4\}$ và thứ tự cần sắp xếp < (tăng dần)

0	1	2	3	4
1	2	3	4	5



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHÈN TRỰC TIẾP

Cài đặt: (trên mảng, tìm tuyến tính kết hợp dời vị trí) Giả sử thứ tự là $<$ (tăng dần)



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHÈN TRỰC TIẾP

```
void insertionSort(int A[], int n) {  
    for (int i = 1; i < n; i++) {  
        int e = A[i]; int k;  
        for (k = i-1; k > -1; k--) {  
            if (A[k] < e) break;  
            A[k+1] = A[k];  
        }  
        A[k+1] = e;  
    }  
}
```



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHÈN TRỰC TIẾP

Cài đặt: (trên danh sách liên kết đơn, tìm tuyến tính) Giả sử thứ tự là $<$ (tăng dần):

Sắp xếp theo nguyên tắc thay đổi liên kết của node



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHÈN TRỰC TIẾP

```
void insertionSort(List &A) {  
    Node *i = A.pHead, *k, *e;  
    while (i->pNext) {  
        q = NULL; k = A.pHead; e=removeAfter(A,i);  
        while (k != i->pNext) {  
            if (!(k->info < e->info)) break;  
            q = k; k = q->pNext;  
        }  
        addAfter(A, e, q);  
        if (i->pNext == e) i = i->pNext;  
    }  
}
```



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHÈN TRỰC TIẾP

Đánh giá:

	TỐT NHẤT (đúng thứ tự)	TRUNG BÌNH (chưa có thứ tự)	XẤU NHẤT (thứ tự ngược)
Theo phép so sánh	$O(n)$	$O(n^2)$	$O(n^2)$
Theo phép gán giá trị khóa	$O(1)$	$O(n^2)$	$O(n^2)$



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CHÈN TRỰC TIẾP

Lưu ý:

- Đối với cấu trúc mảng, thao tác tìm vị trí **k** có thể áp dụng tìm nhị phân → **Binary Insertion Sort**



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP ĐẾM

Từ khóa: Counting Sorting

Điều kiện: Giá trị khóa là số nguyên dương có giá trị lớn nhất không quá lớn.

Phân tích: Nếu giá trị khóa là số nguyên và có thể cấp phát mảng với kích thước bằng giá trị lớn nhất → chỉ cần đếm số lượng giá trị khóa xuất hiện trong danh sách A.



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP ĐẾM

Ý tưởng:

- Giá trị khóa của a_i là chỉ số của mảng **B** có **k** phần tử (**k** là giá trị khóa lớn nhất của **A**)
- Quá trình sắp xếp danh sách **A** là đếm số phần tử của mỗi chỉ số của **B** trong **A**. Từ đó tính ra thứ tự của các khóa a_i trong **A**
- Kết quả sắp xếp có được bằng cách lấy vị trí của phần tử **A** được lưu trong **B**.



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP ĐẾM

Thuật toán:

Đầu vào: $A = \{a_0, a_1, \dots, a_{n-1}\}$ chưa có thứ tự tăng dần

Đầu ra: $C = \{c_0, c_1, \dots, c_{n-1}\}$ đã có thứ tự tăng dần

for $i \leftarrow 0$ to $n-1$ do

$B[A[i]] \leftarrow B[A[i]] + 1$

for $i \leftarrow 1$ to $n-1$ do //tùy thứ tự cần xếp

$B[i] \leftarrow B[i-1] + B[i]$

for $i \leftarrow n-1$ down to 0 do

$B[A[i]] \leftarrow B[A[i]] - 1, C[B[A[i]]] \leftarrow A[i]$



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP ĐẾM

Quá trình tính toán:

Giả sử $A = \{3, 2, 5, 1, 4\}$ và thứ tự cần sắp xếp < (tăng dần)

	0	1	2	3	4
A	3	2	5	1	4

	0	1	2	3	4	5
B	0	0	0	0	0	0

Đếm các
khóa



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP ĐẾM

Quá trình tính toán:

Giả sử $A = \{3, 2, 5, 1, 4\}$ và thứ tự cần sắp xếp < (tăng dần)

	i=0	1	2	3	4
A	3	2	5	1	4

	0	1	2	3	4	5
B	0	0	0	1	0	0



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP ĐẾM

Quá trình tính toán:

Giả sử $A = \{3, 2, 5, 1, 4\}$ và thứ tự cần sắp xếp < (tăng dần)

	0	$i=1$	2	3	4
A	3	2	5	1	4

	0	1	2	3	4	5
B	0	0	1	1	0	0



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP ĐẾM

Quá trình tính toán:

Giả sử $A = \{3, 2, 5, 1, 4\}$ và thứ tự cần sắp xếp < (tăng dần)

	0	1	$i=2$	3	4
A	3	2	5	1	4

	0	1	2	3	4	5
B	0	0	1	1	0	1



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP ĐẾM

Quá trình tính toán:

Giả sử $A = \{3, 2, 5, 1, 4\}$ và thứ tự cần sắp xếp < (tăng dần)

	0	1	2	$i=3$	4
A	3	2	5	1	4

	0	1	2	3	4	5
B	0	1	1	1	0	1



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP ĐẾM

Quá trình tính toán:

Giả sử $A = \{3, 2, 5, 1, 4\}$ và thứ tự cần sắp xếp < (tăng dần)

	0	1	2	3	$i=4$
A	3	2	5	1	4

	0	1	2	3	4	5
B	0	1	1	1	1	1

Tính toán
thứ tự



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP ĐẾM

Quá trình tính toán:

Giả sử $A = \{3, 2, 5, 1, 4\}$ và thứ tự cần sắp xếp < (tăng dần)

	0	1	2	3	4
A	3	2	5	1	4

	0	1	2	3	4	5
B	0	1	2	3	4	5

Ghi kết
quả



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP ĐẾM

Quá trình tính toán:

Giả sử $A = \{3, 2, 5, 1, 4\}$ và thứ tự sắp xếp $<$ (tăng dần)

	0	1	2	3	$i=4$
A	3	2	5	1	4

C				4	
---	--	--	--	---	--

	0	1	2	3	4	5
B	0	1	2	3	$4-1$	5



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP ĐẾM

Quá trình tính toán:

Giả sử $A = \{3, 2, 5, 1, 4\}$ và thứ tự sắp xếp $<$ (tăng dần)

	0	1	2	$i=3$	4
A	3	2	5	1	4

C	1			4	
---	---	--	--	---	--

	0	1	2	3	4	5
B	0	1-1	2	3	3	5



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP ĐẾM

Quá trình tính toán:

Giả sử $A = \{3, 2, 5, 1, 4\}$ và thứ tự sắp xếp $<$ (tăng dần)

	0	1	$i=2$	3	4
A	3	2	5	1	4

C	1			4	5
---	---	--	--	---	---

	0	1	2	3	4	5
B	0	0	2	3	3	$5-1$



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP ĐẾM

Quá trình tính toán:

Giả sử $A = \{3, 2, 5, 1, 4\}$ và thứ tự sắp xếp $<$ (tăng dần)

	0	i=1	2	3	4
A	3	2	5	1	4

C	1	2		4	5
---	---	---	--	---	---

	0	1	2	3	4	5
B	0	0	2 ₋₁	3	3	4



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP ĐẾM

Quá trình tính toán:

Giả sử $A = \{3, 2, 5, 1, 4\}$ và thứ tự sắp xếp $<$ (tăng dần)

	i=0	1	2	3	4
A	3	2	5	1	4

C	1	2	3	4	5
---	---	---	---	---	---

	0	1	2	3	4	5
B	0	0	1	3-1	3	4



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP ĐẾM

Cài đặt: (trên mảng) Giả sử thứ tự cần sắp xếp < (tăng dần)

```
void countingSort(int A[], int n, int C[]) {  
    int B[MAX] = {0}; // đã khai báo MAX  
    for (int i=0; i<n; i++)  
        B[A[i]]++;  
    for (int i=1; i<MAX; i++)  
        B[i] += B[i-1];  
    for (int i=n-1; i>=0; i--)  
        { B[A[i]]--; C[B[A[i]]] = A[i]; }  
}
```



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP ĐẾM

Đánh giá:

Trong mọi trường hợp, độ phức tạp tính toán của Counting Sort là $O(n+k)$, trong đó k là kích thước của mảng B .

Counting Sort là một trong những thuật toán sắp xếp không dựa vào kết quả so sánh giá trị khóa của các phần tử trong danh sách.



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CƠ SỐ

Từ khóa: Radix Sort

Điều kiện: Giá trị khóa là những giá trị rời rạc (số nguyên không âm hoặc chuỗi)

Phân tích Thứ tự của số nguyên dương là do thứ tự của các số trong cơ số theo từng hàng (đơn vị, chục, trăm, ...)



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CƠ SỐ

Ý tưởng:

Cho hai số $a = a_m a_{m-1} \dots a_1$ và $b = b_n b_{n-1} \dots b_1$.

- $a > b$ nếu chữ số tại hàng i cao nhất của hai số thỏa điều kiện: $a_i > b_i$.
- Chữ số ở mỗi hàng chỉ có k giá trị có thứ tự theo cơ số k . Số lớn nhất có m chữ số có tương ứng m hàng. Nếu số có n chữ số ($n < m$) thì chữ số tại hàng cao hơn n là 0.



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CƠ SỐ

Ý tưởng:

- Xét các khóa theo từng hàng, từ hàng thấp nhất đến **m**:
 - Đưa mỗi phần tử vào danh sách **T[j]** tương ứng với giá trị của hàng đang xét, theo thứ tự xét phần tử
 - Nối các danh sách **T[j]** theo thứ tự thích hợp dựa trên thứ tự của các giá trị trong cơ sở.



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CƠ SỐ

Thuật toán: (dùng cho số nguyên dương)

Đầu vào: $A = \{a_0, a_1, \dots, a_{n-1}\}$ chưa có thứ tự

Đầu ra: $A = \{a_0, a_1, \dots, a_{n-1}\}$ đã có thứ tự

$k \leftarrow 10$

while $k \leq 10^m$

for $i \leftarrow 0$ to $n-1$ do

$j \leftarrow (A[i] \bmod k) * 10 \div k$

$T[j] \leftarrow T[j] \cup \{A[i]\}$

$A \leftarrow T[1] \cup T[2] \cup \dots \cup T[m-1]$

$k \leftarrow k * 10$



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CƠ SỞ

Quá trình tính toán:

Giả sử $A = \{312, 142, 151, 1, 40\}$, $m = 3$, $k = 10$ và thứ tự cần sắp xếp $<$ (tăng dần)

$k = 10$, $A = \{312, 142, 151, 1, 40\}$

Xử lý: 312

0	1	2	3	4	5	6	7	8	9
		312							



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CƠ SỐ

Quá trình tính toán:

Giả sử $A = \{312, 142, 151, 1, 40\}$, $m = 3$, $k = 10$ và thứ tự cần sắp xếp $<$ (tăng dần)

$k = 10$, $A = \{312, 142, 151, 1, 40\}$

Xử lý: 412

0	1	2	3	4	5	6	7	8	9
		312							
		412							



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CƠ SỐ

Quá trình tính toán:

Giả sử $A = \{312, 142, 151, 1, 40\}$, $m = 3$, $k = 10$ và thứ tự cần sắp xếp $<$ (tăng dần)

$k = 10$, $A = \{312, 142, 151, 1, 40\}$

Xử lý: 151

0	1	2	3	4	5	6	7	8	9
	151	312							
		412							



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CƠ SỐ

Quá trình tính toán:

Giả sử $A = \{312, 142, 151, 1, 40\}$, $m = 3$, $k = 10$ và thứ tự cần sắp xếp $<$ (tăng dần)

$k = 10$, $A = \{312, 142, 151, 1, 40\}$

Xử lý: 1

0	1	2	3	4	5	6	7	8	9
	151	312							
	1	412							



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CƠ SỐ

Quá trình tính toán:

Giả sử $A = \{312, 142, 151, 1, 40\}$, $m = 3$, $k = 10$ và thứ tự cần sắp xếp $<$ (tăng dần)

$k = 10$, $A = \{312, 142, 151, 1, 40\}$

Xử lý: 40

0	1	2	3	4	5	6	7	8	9
40	151	312							
	1	412							



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CƠ SỐ

Quá trình tính toán:

Giả sử $A = \{312, 142, 151, 1, 40\}$, $m = 3$, $k = 10$ và thứ tự cần sắp xếp $<$ (tăng dần)

$k = 10$, $A = \{312, 142, 151, 1, 40\}$

Nối các danh sách con được $A = \{40, 151, 1, 312, 412\}$

0	1	2	3	4	5	6	7	8	9
40	151	312							
	1	412							



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CƠ SỐ

Quá trình tính toán:

Giả sử $A = \{312, 142, 151, 1, 40\}$, $m = 3$, $k = 10$ và thứ tự cần sắp xếp $<$ (tăng dần)

$k = 100$, $A = \{40, 151, 1, 312, 412\}$

Xử lý 40

0	1	2	3	4	5	6	7	8	9
				40					



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CƠ SỞ

Quá trình tính toán:

Giả sử $A = \{312, 142, 151, 1, 40\}$, $m = 3$, $k = 10$ và thứ tự cần sắp xếp $<$ (tăng dần)

$k = 100$, $A = \{40, 151, 1, 312, 412\}$

Xử lý 151

0	1	2	3	4	5	6	7	8	9
				40	151				



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CƠ SỞ

Quá trình tính toán:

Giả sử $A = \{312, 142, 151, 1, 40\}$, $m = 3$, $k = 10$ và thứ tự cần sắp xếp $<$ (tăng dần)

$k = 100$, $A = \{40, 151, 1, 312, 412\}$

Xử lý 1

0	1	2	3	4	5	6	7	8	9
1				40	151				



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CƠ SỞ

Quá trình tính toán:

Giả sử $A = \{312, 142, 151, 1, 40\}$, $m = 3$, $k = 10$ và thứ tự cần sắp xếp $<$ (tăng dần)

$k = 100$, $A = \{40, 151, 1, 312, 412\}$

Xử lý 312

0	1	2	3	4	5	6	7	8	9
1	312			40	151				



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CƠ SỞ

Quá trình tính toán:

Giả sử $A = \{312, 142, 151, 1, 40\}$, $m = 3$, $k = 10$ và thứ tự cần sắp xếp $<$ (tăng dần)

$k = 100$, $A = \{40, 151, 1, 312, 412\}$

Xử lý 412

0	1	2	3	4	5	6	7	8	9
1	312			40	151				
	412								



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CƠ SỐ

Quá trình tính toán:

Giả sử $A = \{312, 142, 151, 1, 40\}$, $m = 3$, $k = 10$ và thứ tự cần sắp xếp $<$ (tăng dần)

$k = 100$, $A = \{40, 151, 1, 312, 412\}$

Nối các danh sách con được $A = \{1, 312, 412, 40, 151\}$

0	1	2	3	4	5	6	7	8	9
1	312			40	151				
	412								



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CƠ SỐ

Quá trình tính toán:

Giả sử $A = \{312, 142, 151, 1, 40\}$, $m = 3$, $k = 10$ và thứ tự cần sắp xếp $<$ (tăng dần)

$k = 1000$, $A = \{1, 312, 412, 40, 151\}$

Xử lý 1

0	1	2	3	4	5	6	7	8	9
1									



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CƠ SỐ

Quá trình tính toán:

Giả sử $A = \{312, 142, 151, 1, 40\}$, $m = 3$, $k = 10$ và thứ tự cần sắp xếp $<$ (tăng dần)

$k = 1000$, $A = \{1, 312, 412, 40, 151\}$

Xử lý 312

0	1	2	3	4	5	6	7	8	9
1			312						



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CƠ SỐ

Quá trình tính toán:

Giả sử $A = \{312, 142, 151, 1, 40\}$, $m = 3$, $k = 10$ và thứ tự cần sắp xếp $<$ (tăng dần)

$k = 1000$, $A = \{1, 312, 412, 40, 151\}$

Xử lý 412

0	1	2	3	4	5	6	7	8	9
1			312	412					



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CƠ SỐ

Quá trình tính toán:

Giả sử $A = \{312, 142, 151, 1, 40\}$, $m = 3$, $k = 10$ và thứ tự cần sắp xếp $<$ (tăng dần)

$k = 1000$, $A = \{1, 312, 412, 40, 151\}$

Xử lý 40

0	1	2	3	4	5	6	7	8	9
1			312	412					
40									



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CƠ SỞ

Quá trình tính toán:

Giả sử $A = \{312, 142, 151, 1, 40\}$, $m = 3$, $k = 10$ và thứ tự cần sắp xếp $<$ (tăng dần)

$k = 1000$, $A = \{1, 312, 412, 40, 151\}$

Xử lý 151

0	1	2	3	4	5	6	7	8	9
1	151		312	412					
40									



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CƠ SỞ

Quá trình tính toán:

Giả sử $A = \{312, 142, 151, 1, 40\}$, $m = 3$, $k = 10$ và thứ tự cần sắp xếp $<$ (tăng dần)

$k = 1000$, $A = \{1, 312, 412, 40, 151\}$

Nối các danh sách con được $A = \{1, 40, 151, 312, 412\}$

0	1	2	3	4	5	6	7	8	9
1	151		312	412					
40									



III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CƠ SỞ

Cài đặt: (dùng danh sách liên kết) Giả sử sắp xếp danh sách số nguyên dương trong hệ thập phân theo thứ tự $<$



III. CÁC GIẢI THUẬT SẮP XẾP

```
void radixSort(List &A) {  
    List rad[10];  
    for (int i = 0; i < 10; i++)  
        createList(rad[i]);  
    int k = 10;  
    for (int m=0;m<MAX;m++){//MAX: số chữ số  
        while(A.pHead) {  
            Node *p = removeHead(A);  
            addTail(rad[(p->info %k)*10/k], p);  
        }  
        A.pHead = A.pTail = NULL;  
    }
```



III. CÁC GIẢI THUẬT SẮP XẾP

```
for (int i=0; i<10; i++)
    if (rad[i].pHead) {
        if (A.pHead)
            A.pTail->pNext = rad[i].pHead;
        else
            A.pHead = rad[i].pHead;
        A.pTail = rad[i].pTail;
        rad[i].pHead = rad[i].pTail = NULL;
    }
k *= 10;
}
```




III. CÁC GIẢI THUẬT SẮP XẾP

❖ PHƯƠNG PHÁP CƠ SỞ

Đánh giá:

- Radix Sort cũng là một trong những thuật toán sắp xếp không dựa trên kết quả so sánh giá trị khóa giữa các phần tử trong danh sách.
- Độ phức tạp tính toán là $O(m.n)$ trong đó m là số ký tự lớn nhất của một phần tử trong danh sách.
- Radix Sort thích hợp với cấu trúc là danh sách liên kết hơn là dùng cấu trúc mảng



III. CÁC GIẢI THUẬT SẮP XẾP

❖ BÀI TẬP

Trong các thuật toán sắp xếp Selection Sort, Insertion Sort, Counting Sort và Radix Sort, thuật toán nào là sắp xếp ổn định (Stable)? Vì sao?



III. CÁC GIẢI THUẬT SẮP XẾP

❖ BÀI TẬP

Cho mảng $A = \{8, 2, 1, 9, 4, 5, 7, 6, 3\}$. Hãy viết hàm sắp xếp và trình bày từng bước quá trình sắp xếp mảng A theo thứ tự giảm dần ($>$) với thuật toán:

- a) Selection Sort
- b) Insertion Sort



III. CÁC GIẢI THUẬT SẮP XẾP

❖ BÀI TẬP

Cài đặt hàm `hexSort(List &A)` để sắp xếp theo cơ số cho dãy `A` chứa các số thập lục phân. Thứ tự sắp xếp là giảm dần.



III. CÁC GIẢI THUẬT SẮP XẾP

❖ BÀI TẬP

Định nghĩa cấu trúc dữ liệu lưu trữ thông tin máy tính gồm: nhãn hiệu máy, tốc độ xử lý (tính theo GHz) và giá bán. Cài đặt các hàm sau

- a) *sortByName(...)* Sắp xếp danh sách máy theo thứ tự tăng dần đối với nhãn hiệu
- b) *sortBySpeed(...)* Sắp xếp danh sách máy theo thứ tự giảm dần đối với tốc độ xử lý
- c) *sort(...)* Sắp xếp danh sách máy theo thứ tự tăng dần giá bán và trong trường hợp cùng giá thì xếp theo thứ tự giảm dần tốc độ xử lý



III. CÁC GIẢI THUẬT SẮP XẾP

❖ BÀI TẬP

d) *filter(...)* Lọc danh sách các máy tính có giá trong đoạn $[p_1, p_2]$ và tốc độ xử lý trong đoạn $[s_1, s_2]$.