

# Danh sách liên kết

GV. Nguyễn Minh Huy

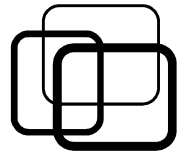


- Khái niệm danh sách liên kết.
- Các thao tác trên danh sách đơn.
- Cải tiến danh sách liên kết đơn.



- **Khái niệm danh sách liên kết.**
- Các thao tác trên danh sách đơn.
- Cải tiến danh sách liên kết đơn.

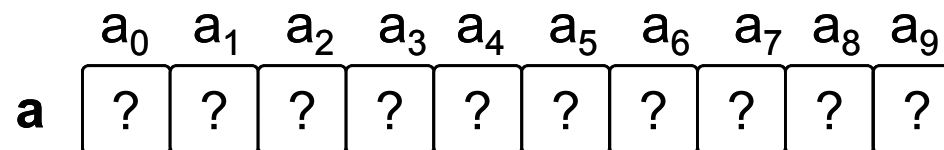
# Khái niệm danh sách liên kết



## ■ Nhận xét về mảng một chiều:

### ■ Tính chất:

- Các phần tử liên tiếp nhau trong bộ nhớ.
- Kích thước bộ nhớ không co dãn.



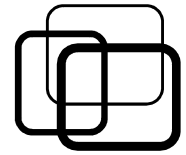
### ■ Ưu điểm:

- Truy xuất phần tử bằng chỉ số  $\rightarrow$  nhanh.
- Thuận tiện lưu số lượng phần tử cố định.

### ■ Khuyết điểm:

- Thêm, xóa phần tử không thuận tiện.
- Thay đổi kích thước phải cấp lại bộ nhớ.
- Tìm vùng nhớ lớn liên tiếp nhau không dễ.

# Khái niệm danh sách liên kết



## ■ Giải pháp danh sách liên kết:

### ■ Bài toán “thuê ngăn tủ đựng đồ”:

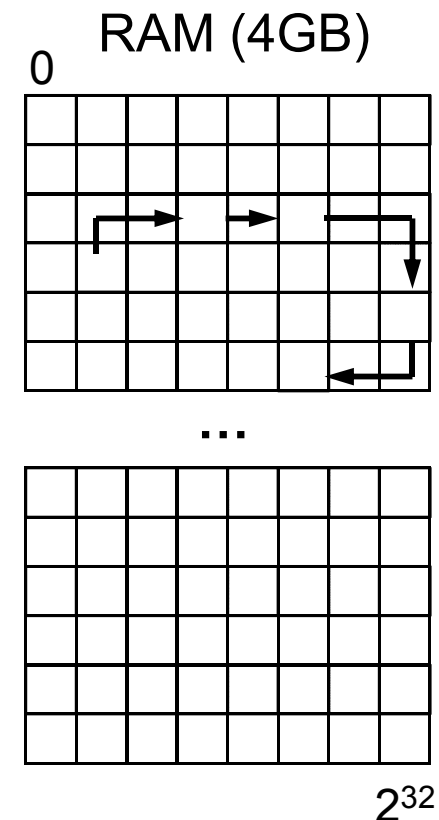
- Cần thuê N ngăn tủ chứa N đồ vật.
- Mỗi ngăn tủ chỉ chứa được 1 đồ vật.

### ■ Giải pháp mảng:

- Thuê N ngăn tủ liên tiếp.
- Giữ STT ngăn đầu tiên.

### ■ Giải pháp danh sách liên kết:

- Thuê N ngăn tủ không liên tiếp.
- Mỗi ngăn chứa:
  - 1 đồ vật.
  - Mẫu giấy ghi STT ngăn tiếp theo.
- Giữ STT ngăn tủ đầu tiên.



# Khái niệm danh sách liên kết



## ■ Danh sách liên kết đơn:

- Một dãy các phần tử không liên tiếp.
- Mỗi phần tử = Dữ liệu + Địa chỉ phần tử kế.
- Phần tử cuối giữ địa chỉ NULL.
- Giữ địa chỉ phần tử đầu.



## ■ Khai báo trong C:

```
struct SNode
{
    int    data;
    SNode *next;
};
```

```
struct SList
{
    SNode *head;
};
```



- Khái niệm danh sách liên kết.
- **Các thao tác trên danh sách đơn.**
- Cải tiến danh sách liên kết đơn.

# Các thao tác trên danh sách đơn

---

## ■ Khởi tạo danh sách:

- Danh sách rỗng.

head●→NULL

## ■ Kiểm tra danh sách rỗng:

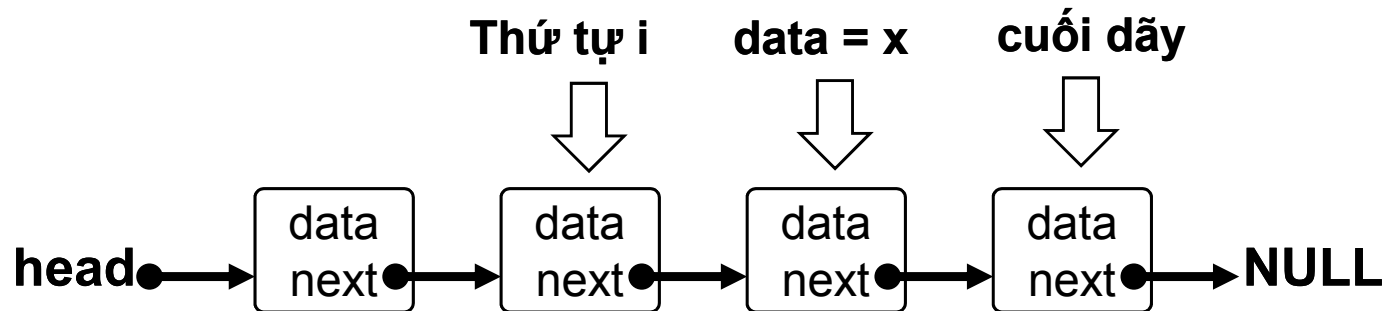
- Kiểm tra head NULL.

head●→NULL ???



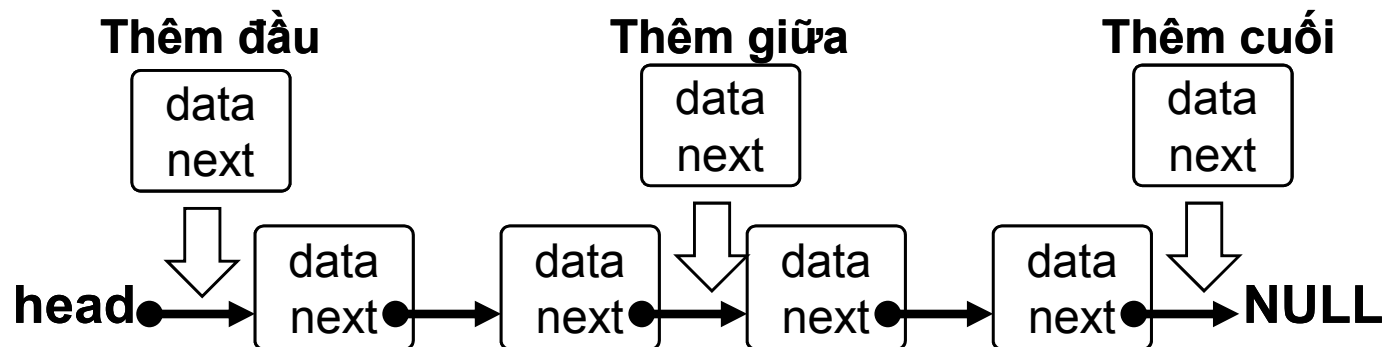
# Các thao tác trên danh sách đơn

- Tìm phần tử:
  - Phần tử thứ  $i$ .
  - Phần tử có dữ liệu  $x$ .
  - Phần tử cuối dãy.



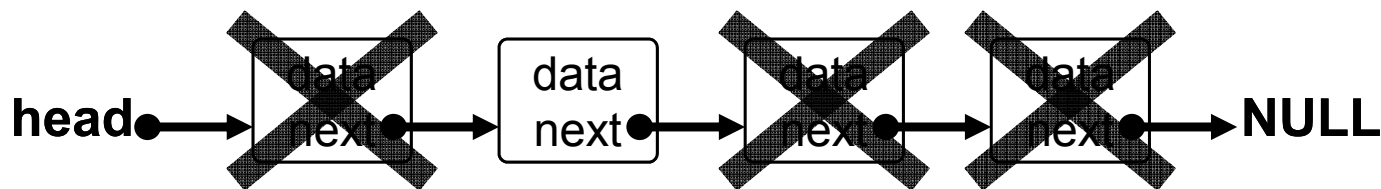
# Các thao tác trên danh sách đơn

- Thêm phần tử:
  - Đầu danh sách.
  - Cuối danh sách.
  - Giữa danh sách:
    - Sau phần tử thứ i.
    - Giữ thứ tự.



# Các thao tác trên danh sách đơn

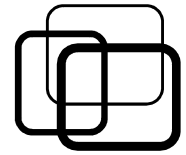
- Xóa phần tử:
  - Đầu danh sách.
  - Cuối danh sách.
  - Giữa danh sách:
    - Sau phần tử i.
    - Phần tử có dữ liệu x.
- Xóa cả danh sách.





- Khái niệm danh sách liên kết.
- Các thao tác trên danh sách đơn.
- **Cải tiến danh sách liên kết đơn.**

# Cải tiến danh sách liên kết đơn



## ■ Danh sách liên kết vs. Mảng một chiều:

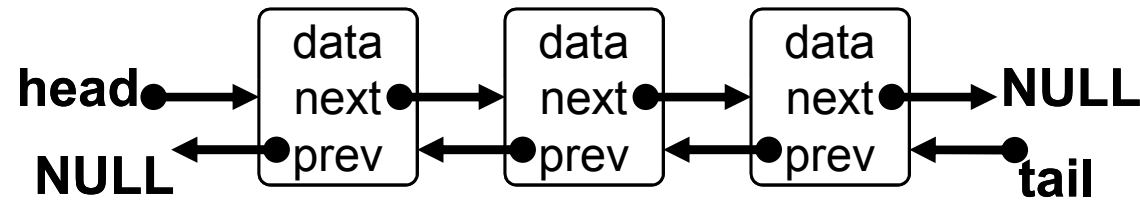
	Danh sách liên kết	Mảng một chiều
Bố trí phần tử	Không liên tục	Liên tục
Giới hạn bộ nhớ	Không giới hạn	Vùng nhớ liên tục
Thay đổi kích thước	Tùy ý Độ phức tạp: <b><math>O(1)</math></b>	Cấp phát lại + sao chép Độ phức tạp: <b><math>O(n)</math></b>
Truy xuất phần tử	Tuần tự Độ phức tạp: <b><math>O(n)</math></b>	Trực tiếp bằng chỉ số. Độ phức tạp: <b><math>O(1)</math></b>
Tìm kiếm	Tiến Độ phức tạp: <b><math>O(n)</math></b>	Tiến và lùi. Độ phức tạp: <b><math>O(n)</math></b>
Thêm/Xóa phần tử	Không dời phần tử Độ phức tạp: <b><math>O(1)</math></b>	Phải dời phần tử Độ phức tạp: <b><math>O(n)</math></b>
Hao phí bộ nhớ	Lưu địa chỉ phần tử kế Hao phí: <b><math>4 * n</math> bytes</b>	Không hao phí

# Cải tiến danh sách liên kết đơn



## ■ Danh sách kép:

- Danh sách đơn duyệt tiến, không lùi.
- Head: duyệt tiến.
- Tail: duyệt lùi.



## ■ Khai báo trong C:

```
struct DNode
{
    int      data;
    DNode   *next;
    DNode   *prev;
};
```

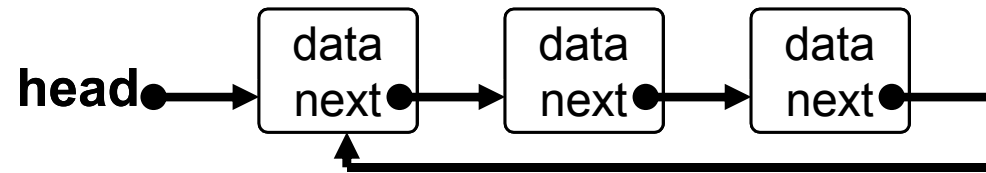
```
struct DList
{
    DNode *head;
    DNode *tail;
};
```

# Cải tiến danh sách liên kết đơn



## ■ Danh sách vòng:

- Cần duyệt danh sách đơn nhiều lần.
- Phần tử danh sách vòng giống danh sách đơn.
- Phần tử cuối trở đến phần tử đầu.



## ■ Khai báo trong C:

```
struct CNode
{
    int      data;
    CNode   *next;
};
```



- Danh sách liên kết:
  - Phần tử lưu trữ phân tán.
  - Phần tử = dữ liệu + địa chỉ phần tử kế.
  - Phần tử cuối trỏ đến NULL.
- Các thao tác trên danh sách đơn:
  - Khởi tạo, kiểm tra rỗng.
  - Tìm, thêm, xóa.
- Cải tiến danh sách liên kết đơn:
  - Danh sách kép.
  - Danh sách vòng.

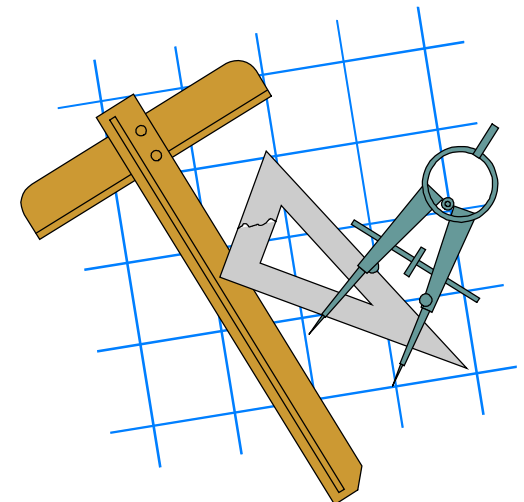






## ■ Bài tập 8.1:

Viết chương trình C cài đặt các thao tác (như trong bài học) trên danh sách kép và danh sách vòng.

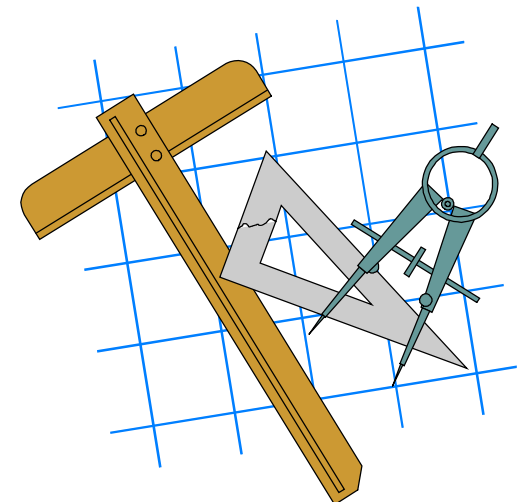




## ■ Bài tập 8.2:

Viết chương trình C cài đặt các thao tác trên danh sách liên kết đơn:

- Đếm số nút trong danh sách.
- Đảo ngược danh sách.
- Thêm phần tử (giữ nguyên thứ tự).





## ■ Bài tập 8.3 (\*):

Một khuyết điểm của danh sách liên kết so với mảng một chiều là khả năng truy xuất phần tử.

Hãy đề xuất một cách cải tiến khả năng truy xuất phần tử của danh sách liên kết sao cho có thể **GẦN BẰNG** với mảng một chiều.

