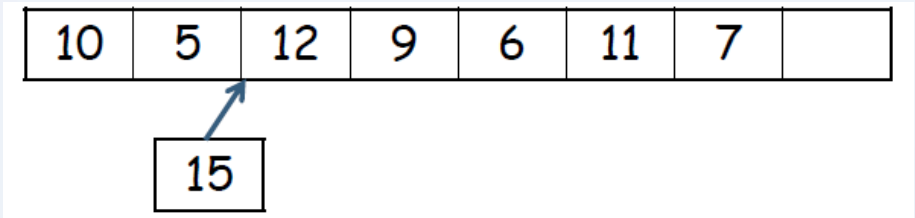
Bài 7

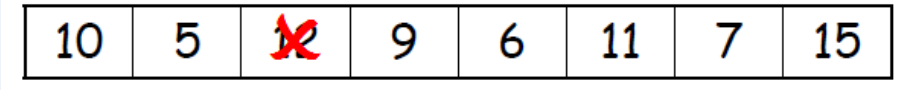
Danh sách liên kết (Linked List)

Vấn đề của Mảng

- Xét lại vấn đề sử dụng mảng để tạo danh sách :
  - Thêm phần tử : O(n)



  - Xoá phần tử : O(n)



  - Số phần tử mảng cố định!!!

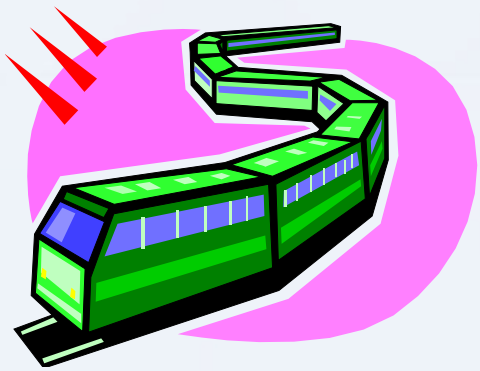
Vấn đề của Mảng

- Làm sao có thể thêm (hay xoá) một phần tử mà không phải di chuyển các phần tủ khác?
- Làm sao để danh sách "động" hơn?
- Cần dùng một cấu trúc lưu trữ mới với các yêu cầu
  - Các phần tử phải được tách rời ra
  - Và được nối với nhau bằng "dây liên kết"
  - Khi thêm phần tử chỉ cần thay đổi mối đây liên kết ⇨ chi phí xử lý sẽ thấp hơn

DANH SÁCH LIÊN KẾT

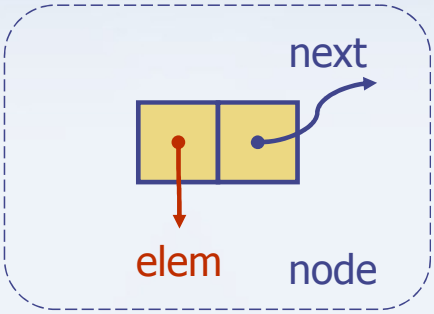
- **Mô hình cấu trúc dữ liệu trừu tượng Linked List là một dãy các vị trí lữu trữ các đối tượng với số lượng tùy ý.**
- **Nó thiết lập một mối quan hệ trước/sau giữa các vị trí**

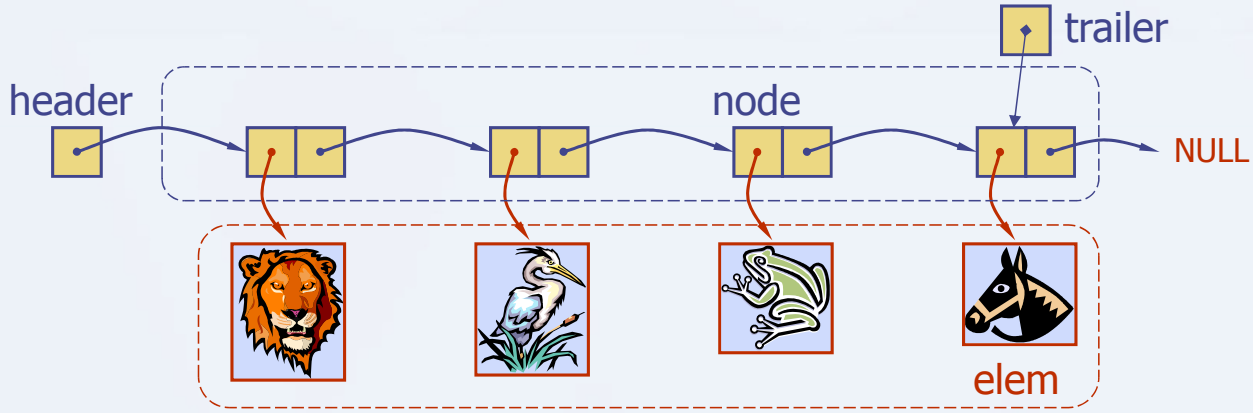
- **Danh sách liên kết đơn**
- **Danh sách liên kết kép**



Danh sách liên kết đơn

- Các nút (node) được cài đặt bao gồm:
  - Phần tử lưu trữ trong nó
  - Một liên kết đến nút kế tiếp
- Sử dụng môt con trỏ header, trỏ vào node đầu danh sách và con trỏ trailer trỏ vào node cuối danh sách.





Cấu trúc của một Node

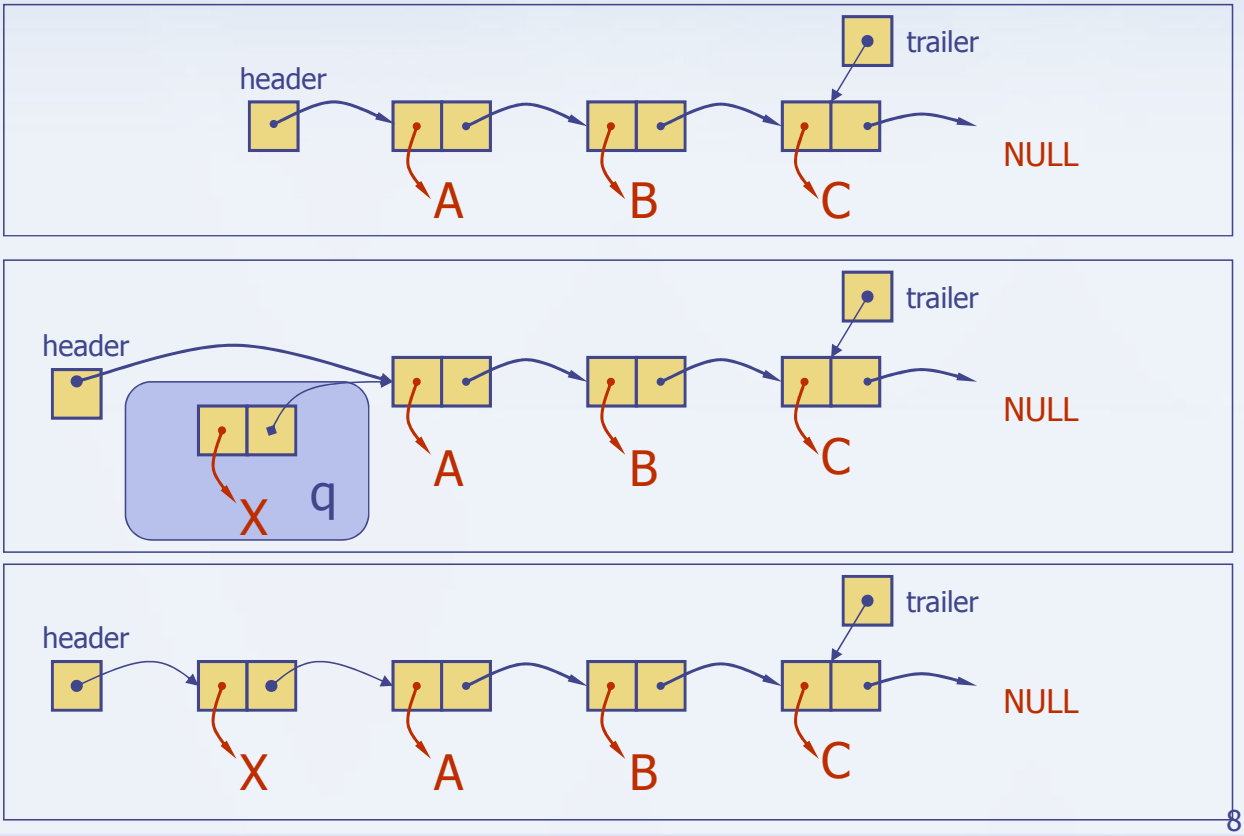
- **Các thuộc tính**
  - **Element *elem;**
  - **Node *next;**
- **Các phương thức**
  - **Node *getnext()** - **Trả lại địa chỉ của nút kế tiếp**
  - **Element *getElem()** - **Trả lại địa chỉ của phần tử mà nút trỏ tới trong nút**
  - **void setNext(Node *)** - **Đặt thuộc tính next trỏ đến đ/c phần tử là đối của phương thức**
  - **void setElem(Element e)** - **Đặt phần tử e vào nút**

Cấu trúc danh sách liên kết đơn

- **Các thuộc tính:**
  - **Node *header**
  - **Node *trailer**
- **Các phương thức chung:**
  - long size(),
  - int isEmpty()
- **Các phương thức truy cập:**
  - Node *first()
  - Node *last()
- **Các phương thức cập nhật:**
  - void replace(Node *p, Element e)
  - Node *insertAfter(Node *p, Element e)
  - Node * insertFirst(Element e)
  - Node * insertLast(Element e)
  - Node * getNode(int i)
  - void remove(Node *p)

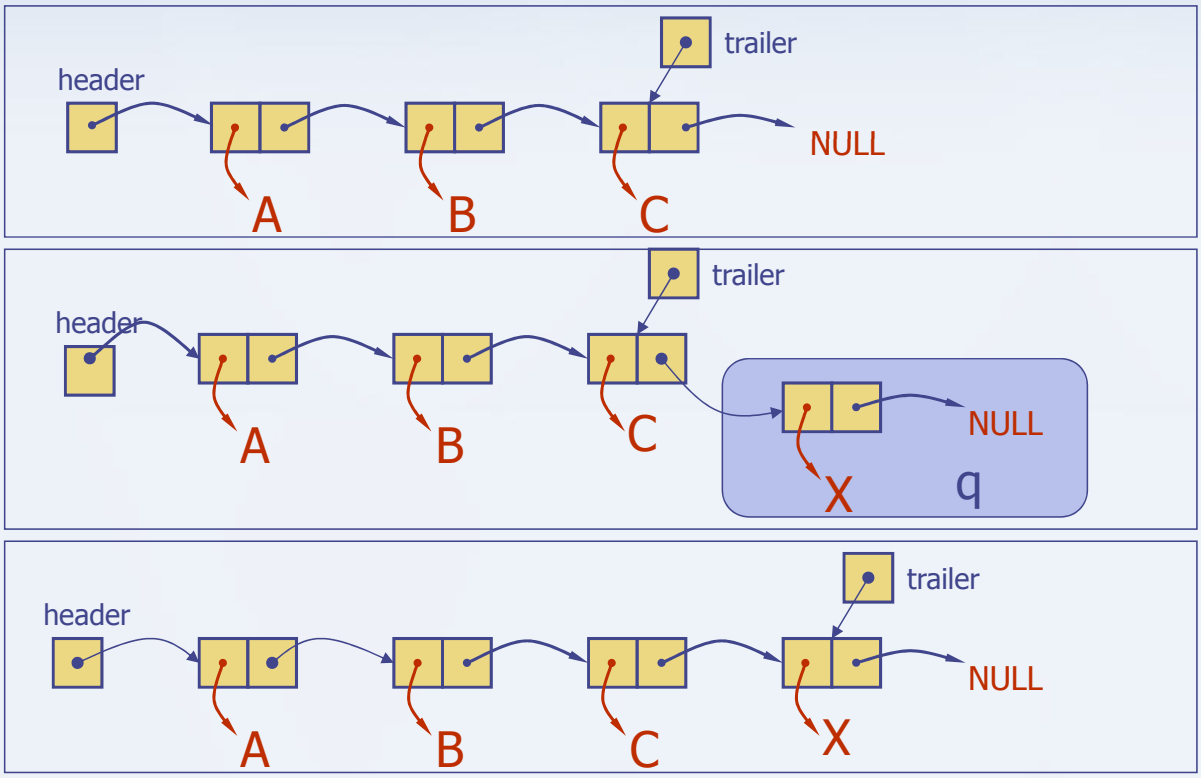
Insertion First

- Hình ảnh phép toán insertFirst(), phép toán trả lại vị trí q



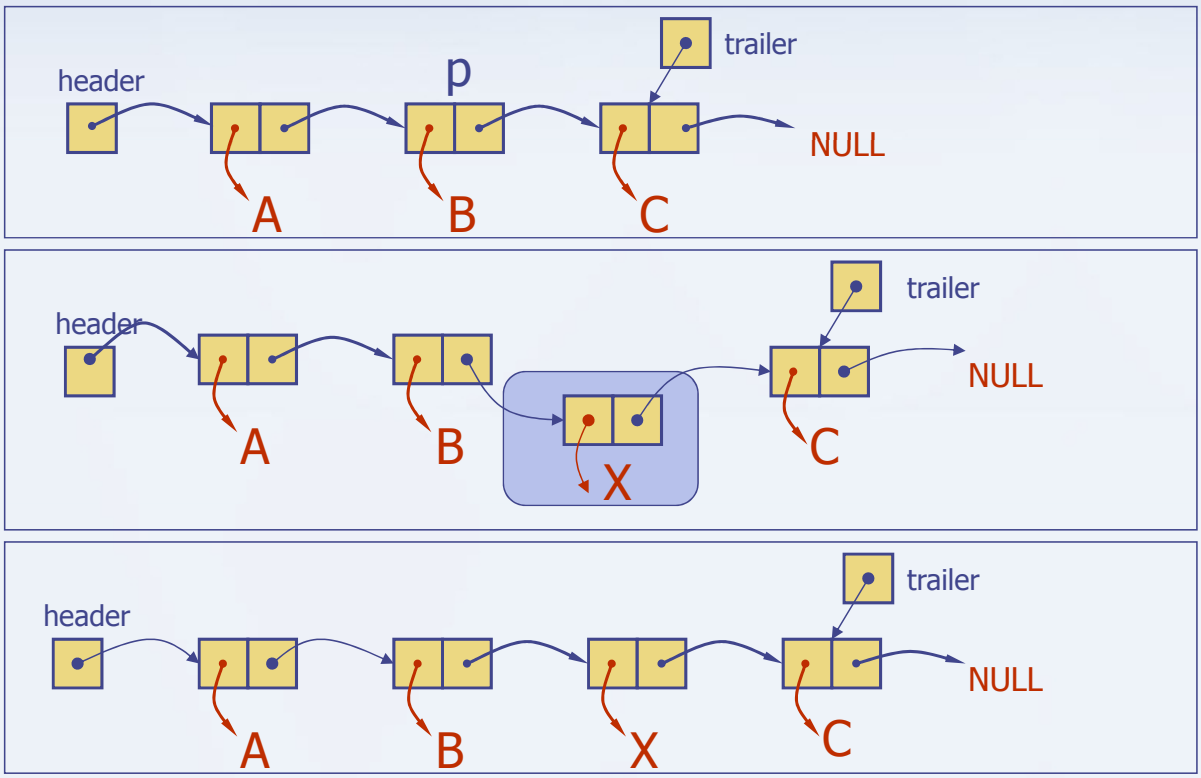
Insertion Last

- Hình ảnh phép toán insertLast(), phép toán trả lại vị trí q



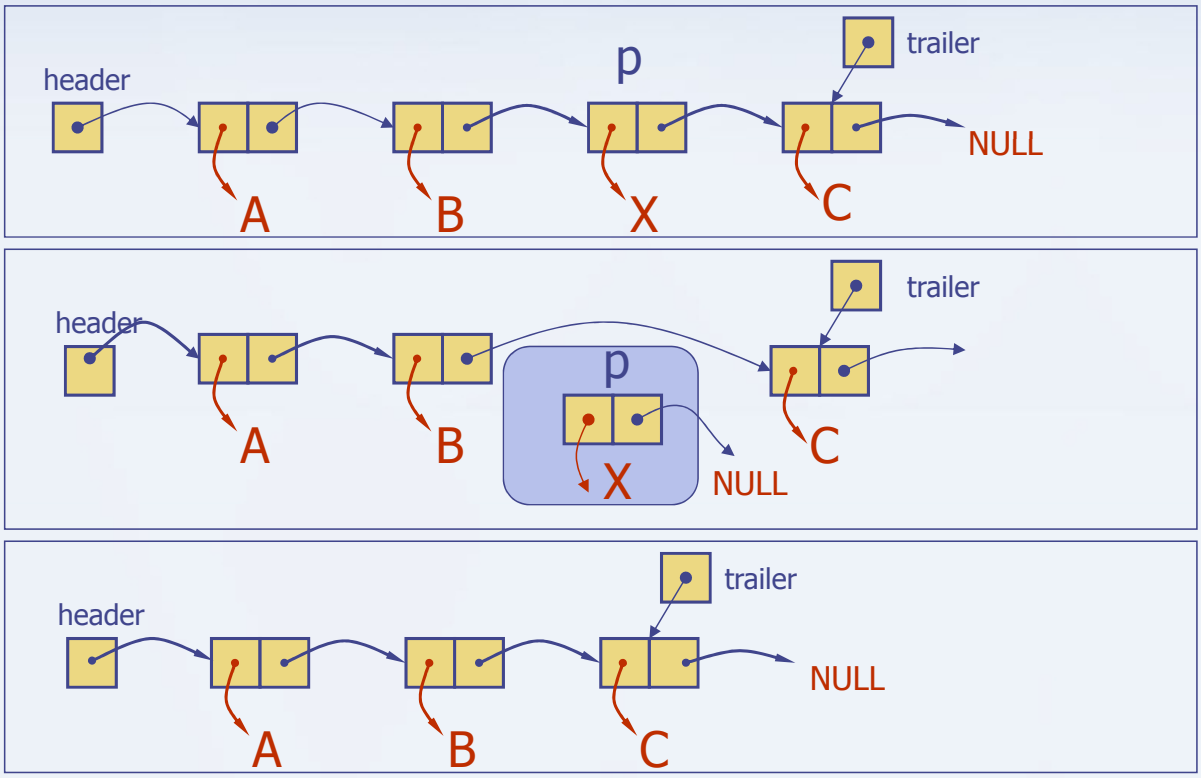
Insertion After

- Hình ảnh phép toán insertAfter(p, X), phép toán trả lại vị trí q



Remove

- Hình ảnh phép toán remove(p)



Bài tập về nhà

Xây dựng lớp ứng dụng sử dụng lớp Danh sách liên kết đơn để lưu trữ 1 danh sách sinh viên. Mỗi sinh viên gồm các thông tin sau: MaSv, Hoten, Ngay, Thang, Nam sinh, gioi tinh, que quan.

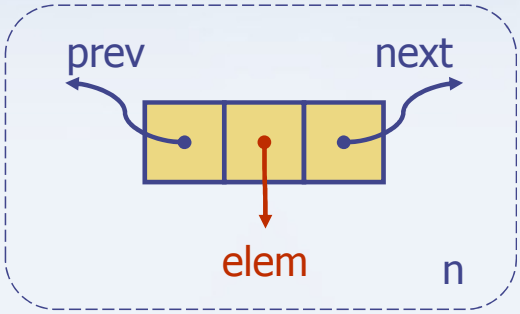
Lớp có các chức năng sau:

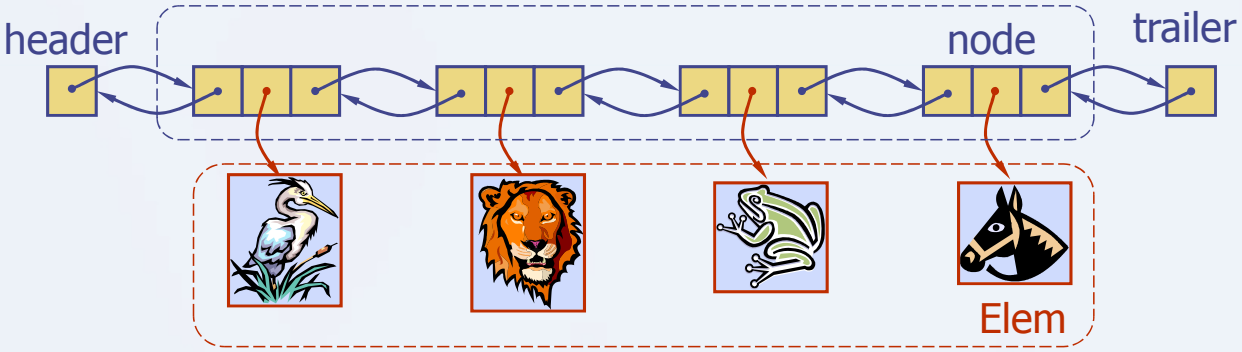
- Thêm một sinh viên vào cuối DS
- Thêm một sinh viên vào đầu DS
- Xóa bỏ một sinh viên thứ i khỏi DS
- Thay thế sinh viên thứ i bằng một sinh viên mới

Xây dựng chương trình để chạy lớp ứng dụng

Danh sách liên kết kép

- Các nút (node) được cài đặt bao gồm:
  - Phần tử lưu trữ trong nó
  - Một liên kết đến nút trước nó
  - Một liên kết đến nút kế tiếp
- Có hai nút đặc biệt là trailer và header





Cấu trúc của một Node

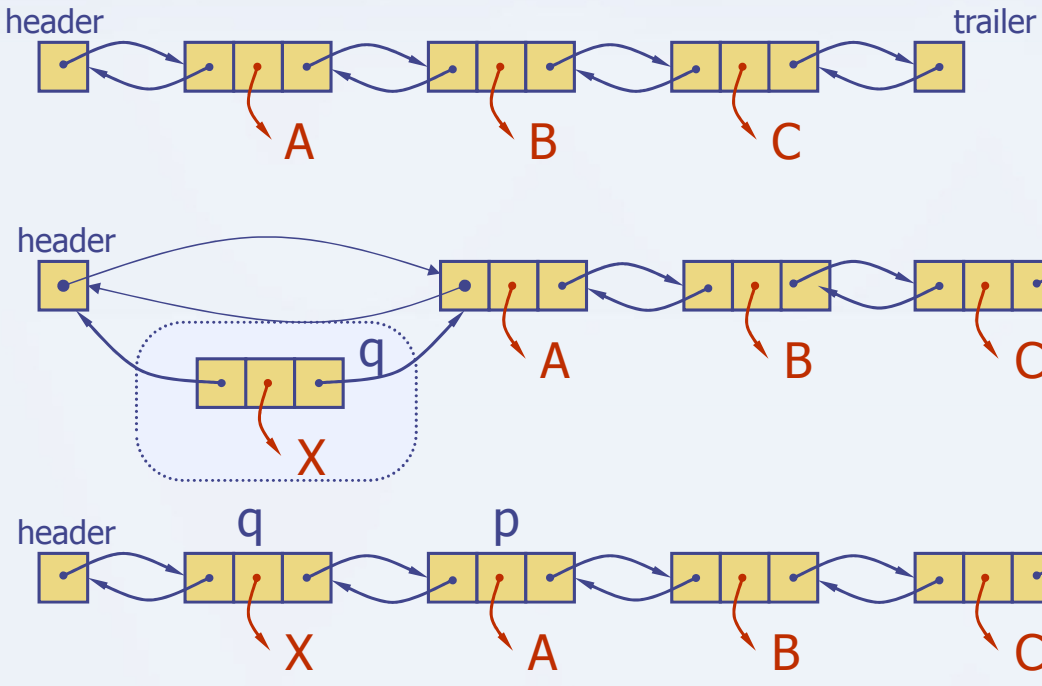
- Các thuộc tính
  - Element *elem;
  - Node *next, *pre;
- Các phương thức
  - Node *getnext() - Trả lại địa chỉ của nút kế tiếp
  - Node *getPre() - Trả lại địa chỉ của nút trước đó
  - Element *getElem() - Trả lại địa chỉ của phần tử lưu trong nút
  - void setNext(Node *) - Đặt thuộc tính Next trỏ đến đ/c của phần tử là đối của phương thức
  - void setPre(Node *) - Đặt thuộc tính Prior trỏ đến đ/c của phần tử là đối của phương thức
  - void setElem(Element e) - Đặt phần tử e vào nút

Cấu trúc Danh sách liên kết kép

- **Các thuộc tính:**
  - **Node *header**
  - **Node *trailer**
- **Các phương thức chung:**
  - long size(),
  - int isEmpty()
- **Các phương thức truy cập:**
  - Node *first()
  - Node *last()
- **Các phương thức cập nhật:**
  - void replace(Node *p, e)
  - Node *insertAfter(Node *p, Elemnt e)
  - Node *insertBefore(Node *p, Element e)
  - Node * insertFirst(Element e)
  - Node * insertLast(Element e)
  - Node * getNode(int i)
  - void remove(Node *p)

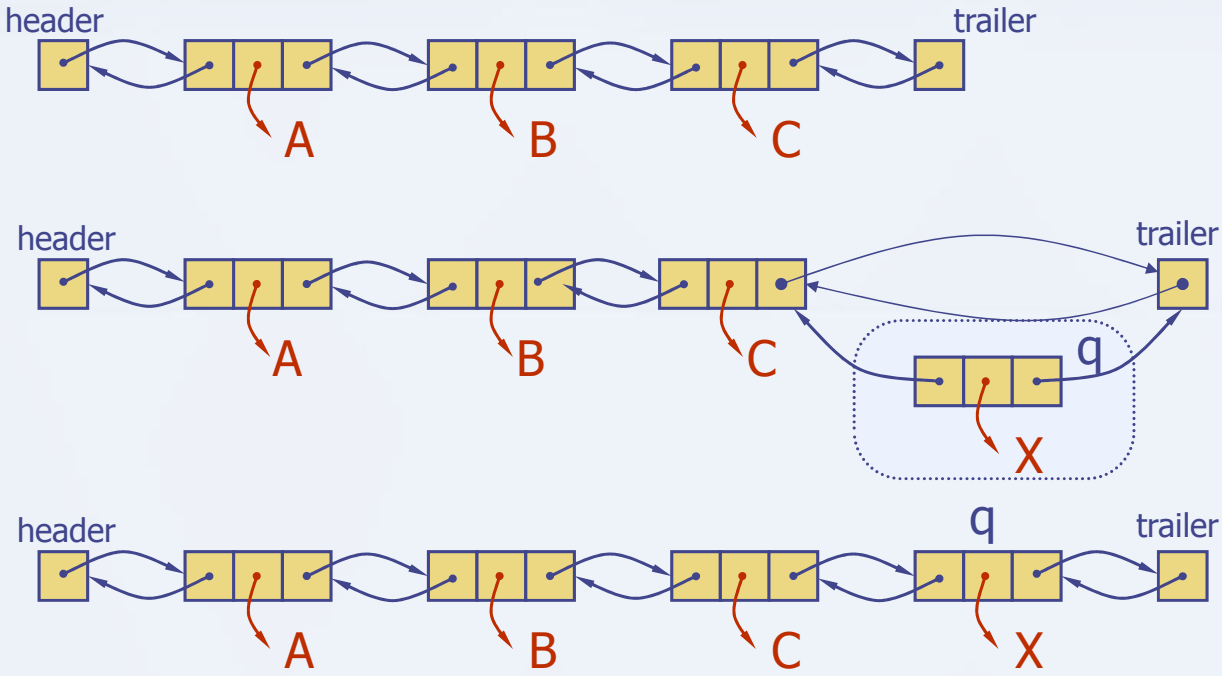
Insert First

- Hình ảnh phép toán insertFirst(X), phép toán trả lại vị trí q



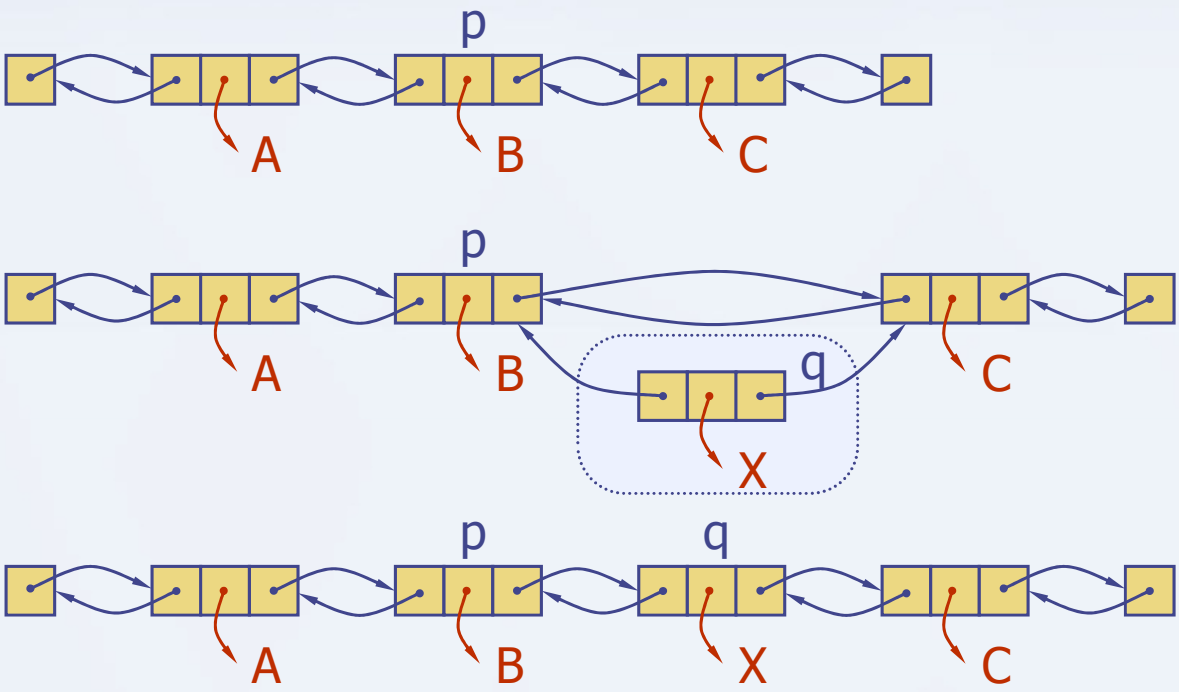
Insert Last

- Hình ảnh phép toán insertLast(X), phép toán trả lại vị trí q



Insert After

- Hình ảnh phép toán insertAfter(p, X), phép toán trả lại vị trí q



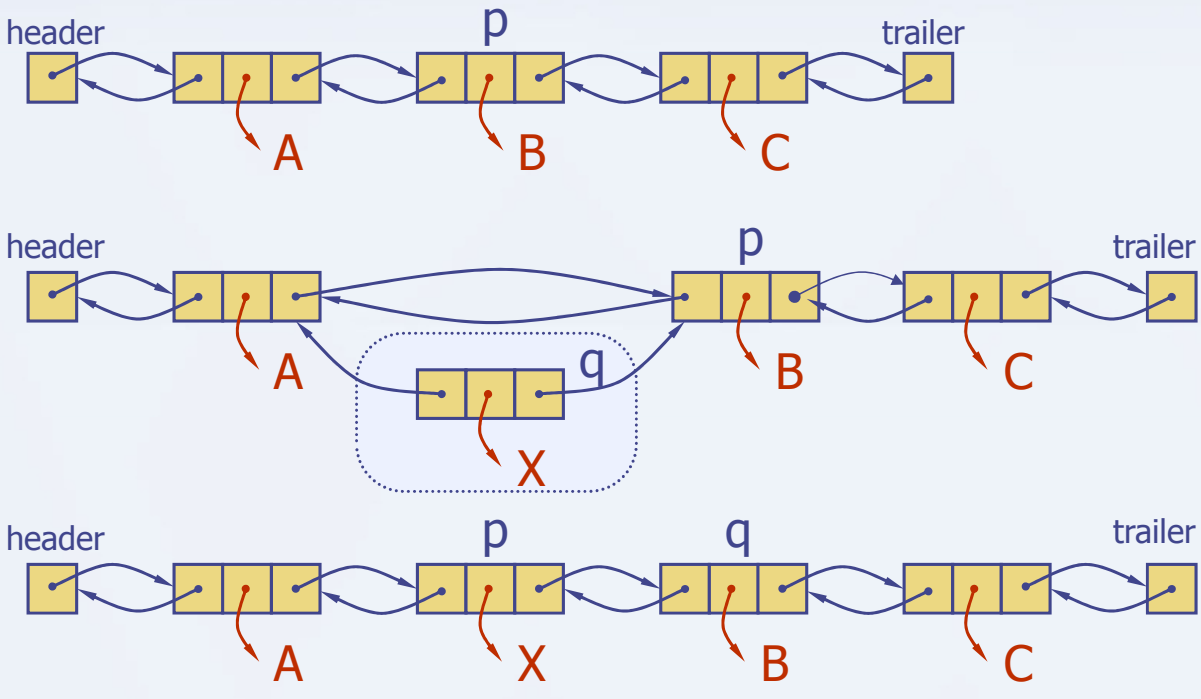
Thuật toán Insert After

Algorithm insertAfter(*p*,*e*): //Bổ sung phần tử e vào sau
// phần tử nút p

Tạo ra một nút mới q
q->setElement(*e*) //Đặt gia trị e vào nút q
q->setNext(p->getNext())//liên kết với phần tử sau nó
p.getNext()->setPrev(q)//Liên kết phần tử sau p với q
q->setPrev(p) //liên kết q với phần tử trước nó
p->setNext(q) //liên kết p với q
return q //trả lại vị trí của q

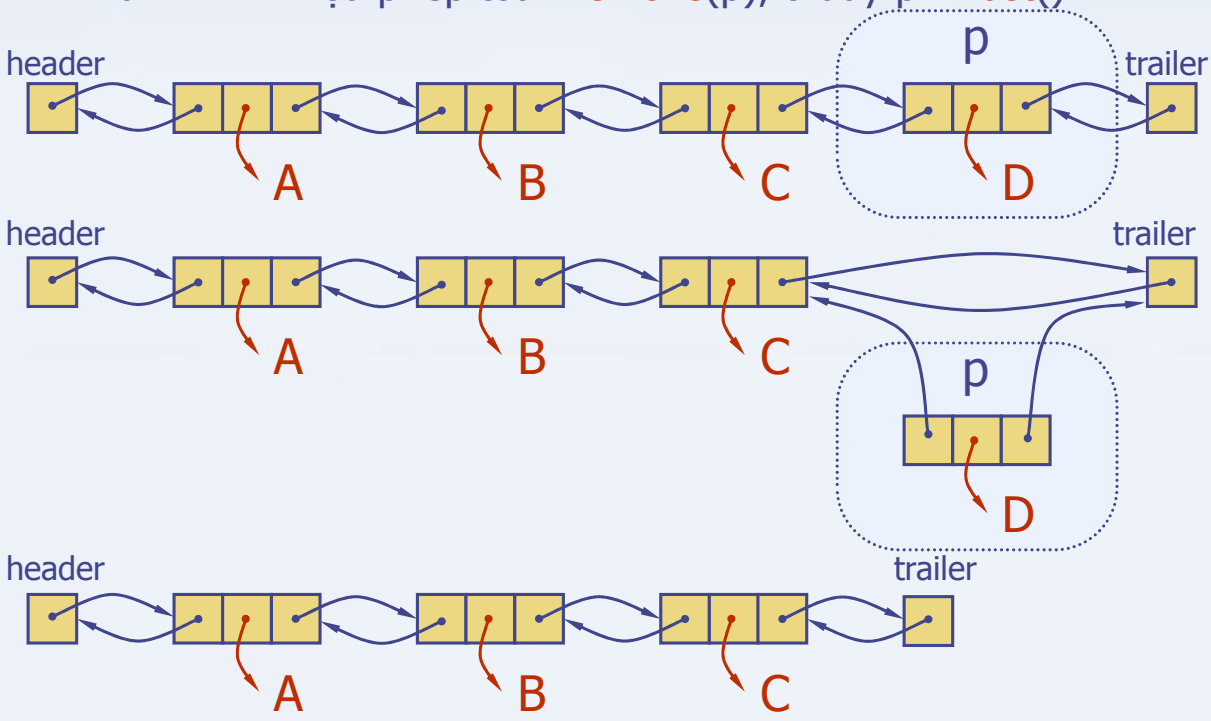
Insert Before

- Hình ảnh phép toán insertBefore(p, X), phép toán trả lại vị trí q



Xóa - Remove

- Hình ảnh minh họa phép toán remove(p), ở đây p = last()



Thuật toán remove

Algorithm remove(Node *p):

```
//kết nối phần tử trước p với phần tử sau p
    p->getPre()->setNext(p->getNext())
//kết nối phần tử sau p với pần tử trước p
    p->getNext()->setPre(p->getPre())
//bỏ kết nối p với phần tử trước nó
    p.setPre(NULL)
    p.setNext(NULL)
    delete p
```

So sánh mảng và DSLK

Mảng

- Bộ nhớ sử dụng lưu trữ phụ thuộc vào việc cài đặt chứ không phải số lượng thực sự cần lưu.
- Mối quan hệ giữa phần tử đầu và các phần tử khác là rất ít
- Các phần tử được sắp xếp cho phép tìm kiếm rất nhanh
- Việc chèn và xóa phần tử đòi hỏi phải di chuyển các phần tử.

Danh sách liên kết

- Bộ nhớ sử dụng để lưu trữ tương ứng với số lượng các phần tử thực sự cần lưu tại bất kỳ thời điểm nào.
- Sử dụng một con trỏ để lưu phần tử đầu, từ đó đi đến các phần tử khác.
- Việc bổ sung và xóa bỏ các phần tử không phải di chuyển các phần tử
- Truy nhập đến các phần tử chỉ có thể thực hiện được bằng cách đi dọc theo chuỗi mắt xích từ phần tử đầu. Vì vậy đối với danh sách liên kết đơn thì thời gian tìm kiếm một phần tử sẽ là $O(n)$.

Bài tập: 17h00 11/11/2015

- Xây dựng lớp Node
- Xây dựng lớp DblList
- Xây dựng lớp DblItr //Lớp bộ lặp
- Xây dựng lớp ứng dụng sử dụng lớp Danh sách liên kết đôi để lưu trữ 1 danh sách sinh viên. Mỗi sinh viên gồm các thông tin sau: MaSv, Hoten, Ngay, Thang, Nam sinh, gioi tinh, que quan.

Lớp có các các chức năng sau:

- Thêm một sinh viên vào cuối DS
- Thêm một sinh viên vào đầu DS
- Xóa bỏ sinh viên thứ i khỏi DS
- Thay thế sinh viên thứ i bằng một sinh viên mới

Xây dựng chương trình để chạy lớp ứng dụng

Hết