Các cấu trúc dữ liệu cơ bản

Giảng viên: Tạ Việt Cường
Phòng HMI – Khoa CNTT

Ôn tập

- Đọc hiểu pseudo-code đơn giản
- Tính được độ phức tạp cơ bản
  - Sắp xếp theo độ phức tạp tăng dần
    1. O(N)
    2. O(NlogN)
    3. O($\sqrt{N}$)
    4. O(N^2)
    5. O(logN)
- Học thuộc 2 thuật toán sắp xếp
  - Sắp xếp lựa chọn
  - Sắp xếp chèn

Bài 5

- Đếm số lần xuất hiện nhiều nhất của 1 số trong dãy số
- Cách đơn giản:
  - Input: N và dãy A[0], A[1],…,A[N-1]
  - Output: Số xuất hiện nhiều nhất và số lần xuất hiện
  - Pseudocode:

```
1  countmax = 0
2  amax = -1
3  for i = 0 -> N-1 do
4      count = 1
5      for j = i +1 -> N-1 do
6          if A[i] == A[j] then
7              count++
8      if (count>countmax)||
   ((count==countmax)&&(A[i]<amax))then
9          countmax = count, amax = A[i]
10 return amax, countmax
```

Bài 7

- Đếm số lần xuất hiện nhiều nhất của 1 số trong dãy số sắp xếp tăng dần
  - Pseudocode:

```
1  countmax = 1
2  amax = A[0]
3  ~~for i = 0 -> N-1 do~~ ***i = 0; while i < N-1 do***
4      count = 1
5      for j = i +1 -> N-1 do
6          if A[i] == A[j] then
7              count++
       ***(**) else break***
8      if (count>countmax)||
   ((count==countmax)&&(A[i]<amax))then
9          countmax = count, amax = A[i]
       ***i = j***
10 return amax, countmax
```

Nội dung tóm tắt

- Mảng Easy
  - Array, Vector
  - 1 chiều, 2 chiều, ...
- Danh sách liên kết Easy
  - Linked-list

- Stack Not so difficult

- Queue Not so difficult

Bài toán thực tế

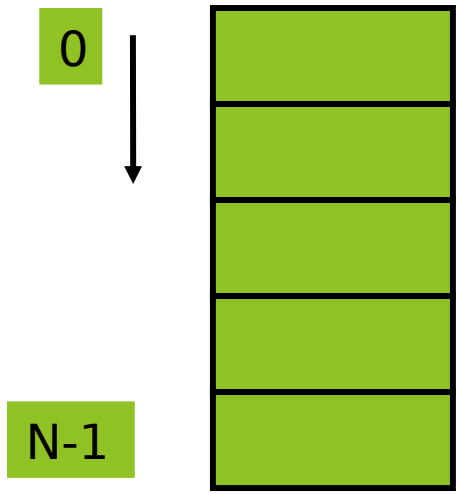
- Bài toán thực tế được biểu diễn dưới dạng các câu truy vấn trên kiểu dữ liệu động
- Ví dụ về một danh sách động:
  - Bài 5:
    - Xóa A[j] nếu A[i] == A[j]
    - Nhanh hay chậm hơn?
  - Dãy người xếp hàng trả tiền ở quầy siêu thị
    - Có nhiều quầy xếp hàng
    - Người đến trước thì được trả tiền trước
    - Có 1 số lối ra, bạn chọn lối ra nào ?
  - Danh sách bạn bè đang online trên facebook
    - Người được chat gần nhất được đưa lên đầu

Phép toán cơ bản trên cấu trúc dữ liệu

- Cho một cấu trúc dữ liệu (mảng, linkedlist, stack, queue...)
- Một số phép toán cơ bản:
  - Thêm 1 phần tử vào danh sách
  - Xóa 1 phần tử khỏi danh sách
  - Tìm kiếm 1 phần tử trong danh sách
  - Tìm phần tử lớn nhất/nhỏ nhất (tùy trường hợp)

Mảng 1 chiều

- Hai cách:
  - Cách 1: int N; int A[N];
  - Cách 2: vector<int> A;
- Độ phức tạp:
  - Thêm 1 phần tử: O(1)
  - Thêm 1 phần tử vào vị trí k: O(N)
  - Xóa 1 phần tử: O(N)
  - Tìm kiếm 1 phần tử: O(N)
  - Tìm phần tử lớn nhất: O(N)



Làm quen với kiểu dữ liệu tổng quát

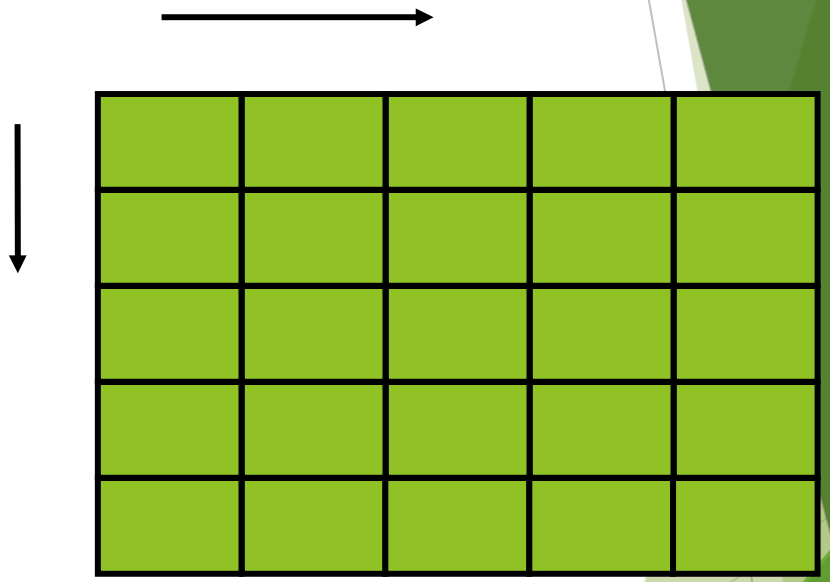
- Dữ liệu trên thực tế thường không biểu diễn dưới dạng int
  - Sinhvien: tên, ngày tháng năm sinh, email, …
  - Gmail
  - Facebook
- Khái niệm về bản ghi (record/struct/class):
  - ID – mã số định danh
  - Tập hợp các dữ liệu khác
    - Trong thực tế rất phức tạp
    - Ví dụ: 1 sinh viên thì cần những thông tin gì

Làm quen với kiểu dữ liệu tổng quát

- Để so sánh 2 sinh viên giống nhau:
  - Dùng MSSV
- Để sắp xếp:
  - Phụ thuộc vào điều kiện muốn sắp xếp
    - Theo tên
    - Theo điểm
  - Sử dụng hàm compare(sinhvien1, sinhvien2)
    - Có thể thay đổi mục tiêu bằng cách truy cập vào các giá trị

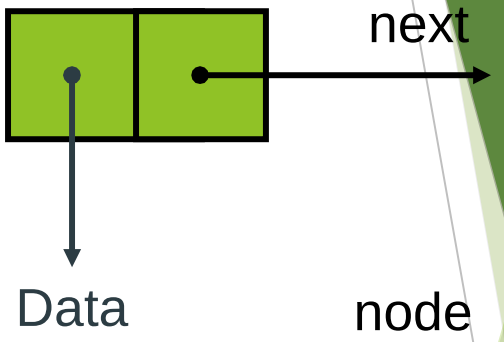
Mảng 2 chiều

- 2 cách khai báo:
  - **datatype** A[100][100];
  - vector<vector<**datatype**> > A;
- Bài tập:
  - Cho mảng 2 chiều A(MxN) gồm các phần tử giá trị 0 hoặc 1
  - Tìm hình chữ nhật toàn 0 có diện tích lớn nhất

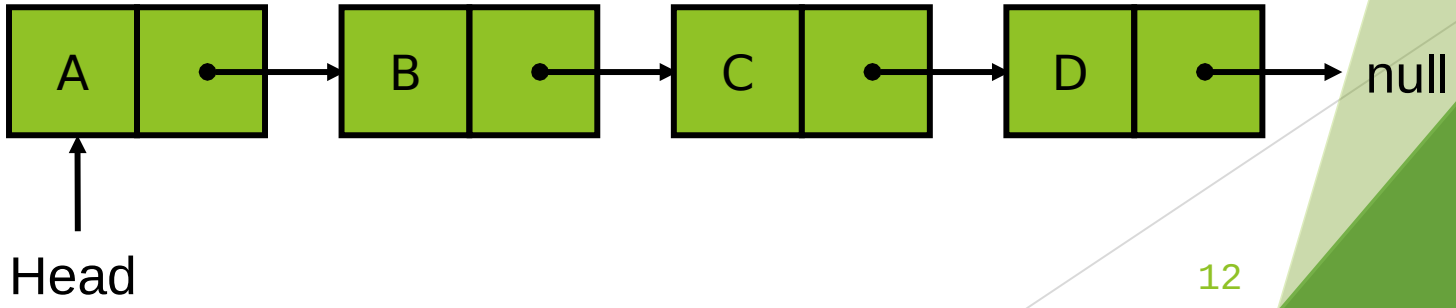


Danh sách liên kết đơn

- Phần tử của danh sách gồm 2 phần
  - Dữ liệu Data (thông tin về sinh viên)
  - Con trỏ next



- Danh sách liên kết: chuỗi các phần tử



Danh sách liên kết đơn

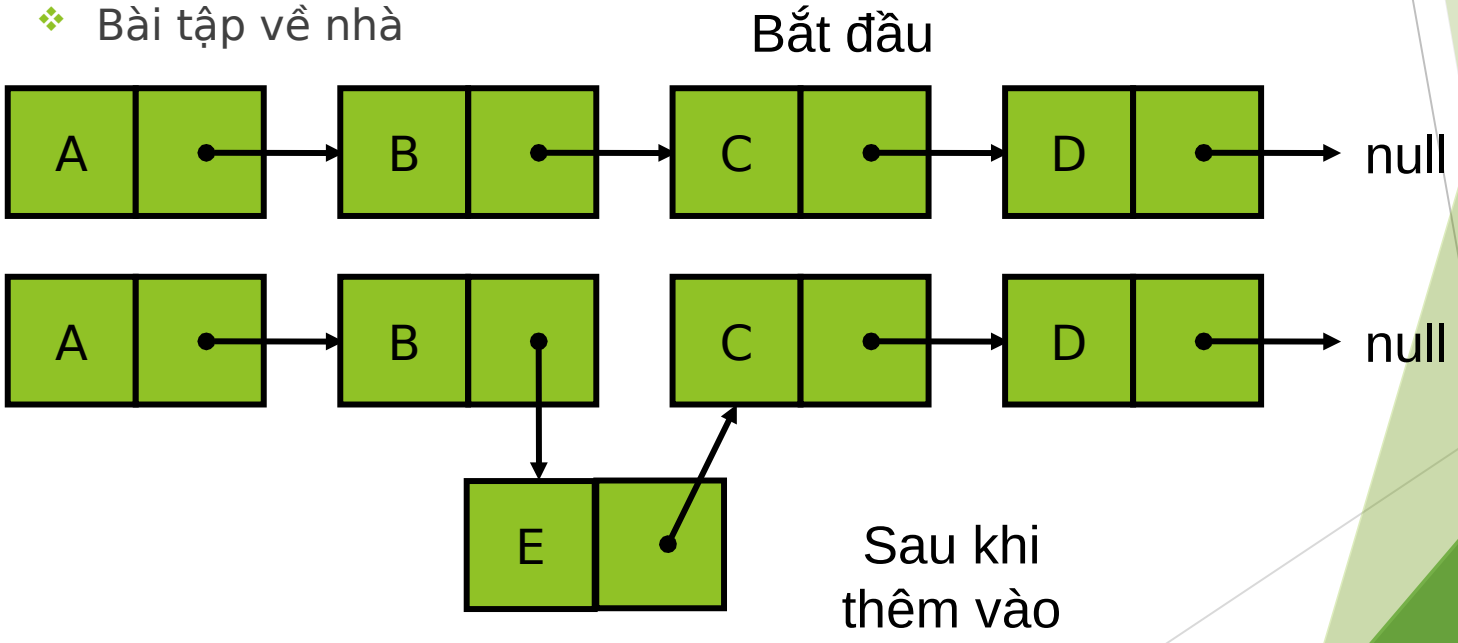
- Duyệt/in ra cả danh sách

```
ptr = head
while (ptr != null) do
    in ra ptr->data
    ptr = ptr-> next
```

- Tìm kiếm 1 phần tử giá trị là x
  - Thay lệnh in ra = lệnh so sánh

Danh sách liên kết đơn

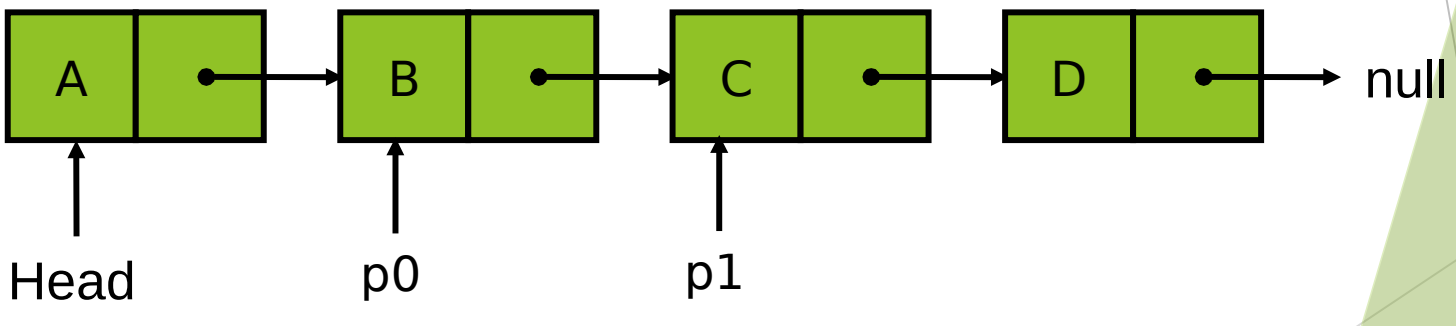
- Thêm vào 1 phần tử tại vị trí thứ k (head = 0)
  - Step 1: Tìm ví trí thứ k
  - Step 2: Tạo phần tử mới có giá trị là data là E
  - Step 3: Thay đổi giá trị con trỏ
  - Bài tập về nhà



Số lượng con trỏ phải thay đổi = ?

Danh sách liên kết đơn

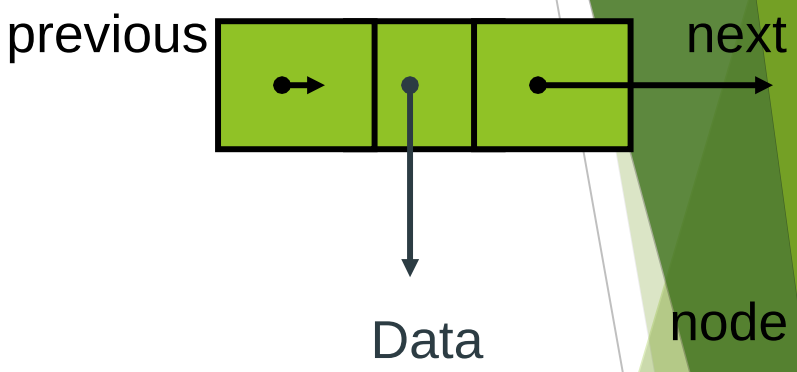
- Xóa 1 phần tử tại vị trí thứ k
  - Step 0: Sử dụng 2 con trỏ p0, p1
  - Step 1: Tìm vị trí thứ k, ở p1 (có p1 = p0->next)
  - Step 2: Sửa lại p0->next = p1->next

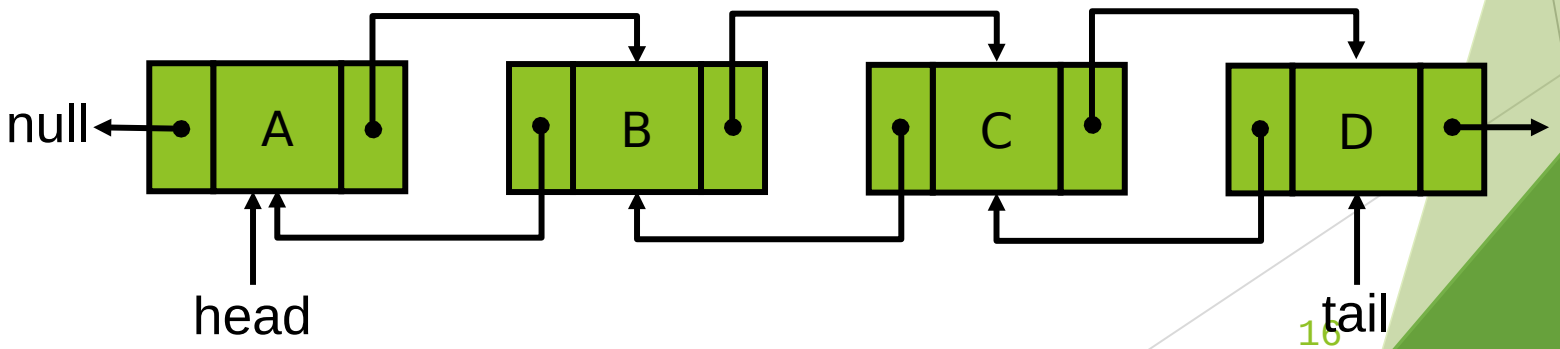


- Số lượng con trỏ phải thay đổi = ?

Danh sách liên kết kép

- Giống với danh sách đơn, ngoại trừ
  - Thêm vào 1 con trỏ previous



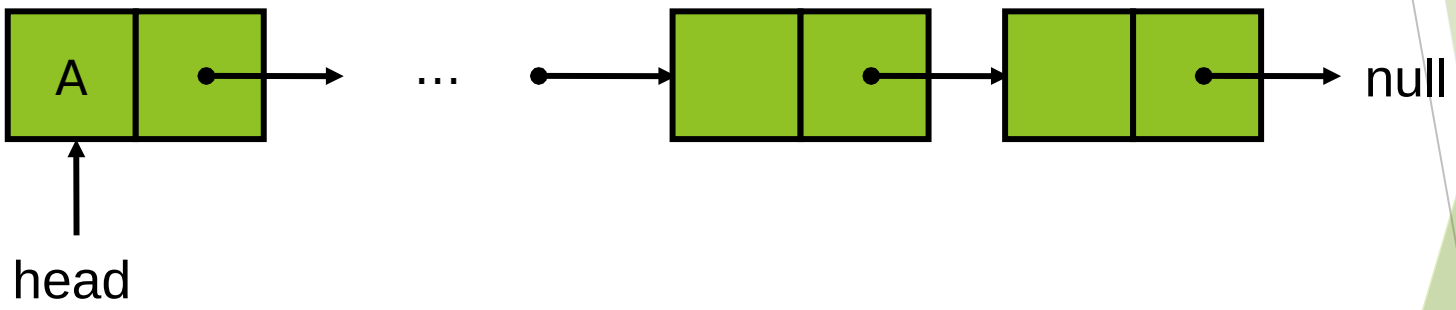


Danh sách liên kết kép

- Điểm khác biệt
  - Di chuyển theo 2 chiều
- Trade-off: tốn nhiều bộ nhớ để lưu các con trỏ (gấp đôi)
- Độ phức tạp:
  - Tương đương

Bài tập ví dụ: Cẩn thận các trường hợp của input

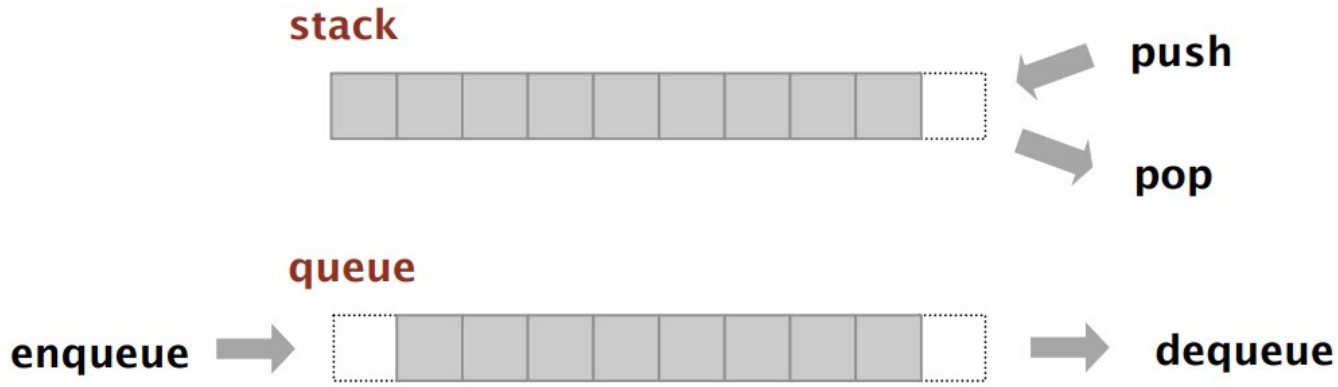
- Cho danh sách liên kết được trỏ bởi con trỏ head như sau:



- Viết pseudo-code của insert(X, k): chèn vào vị trí thứ k giá trị X, với vị trí đc trỏ bởi head là vị trí thứ 1

Stack và Queue

- Dữ liệu được lấy ra theo “thời điểm” xuất hiện trong danh sách:
  - Stack: Last in First Out
  - Queue: First In First Out
- 2 phép toán cơ bản:
  - **push/enqueue**: thêm phần tử vào
  - **pop/dequeue**: lấy phần tử ra, và trả về giá trị của phần tử đó



Kiểu dữ liệu stack (ngăn xếp)

- Lấy ra phần tử vừa được thêm vào
  - Dùng phổ biến trong cấu trúc tổ chức bộ nhớ của máy tính
- Ví dụ đơn giản: Cài đặt stack bằng mảng

Stack mở rộng về hướng

Phép toán	N	Stack	Return
push 1	1	1	
push 2	2	1 2	
push 5	3	1 2 5	
pop	2	1 2	5
push 1	3	1 2 1	
pop	2	1 2	1
pop	1	1	2

Kiểu dữ liệu queue (hàng đợi)

- Lấy ra phần tử được thêm vào đầu tiên (sớm nhất)
  - Xếp hang
- Ví dụ đơn giản: Cài đặt queue bằng mảng

Queue mở rộng về hướng

Phép toán	N	Queue	Return
push 1	1	1	
push 2	2	1 2	
push 5	3	1 2 5	
pop	2	_ 2 5	1
push 1	3	_ 2 5 1	
pop	2	_ _ 5 1	2
pop	1	_ _ _ 1	5

Các kiểu bài tập có thể có về queue và stack

- Cài đặt bằng array, linked-list, double linked-list
  - Viết pseudo-code cho các hàm *push, pop*
  - Xử lí các trường hợp đặc biệt:
    - *pop* khi stack/queue bị rỗng
- Cho trạng thái stack/queue + dãy các phép toán:
  - In ra stack/queue sau khi thực hiện
  - In ra kết quả của phép toán *pop*
  - Ví dụ:
    - Stack: [push 2, push 1, pop, push 2, push 3, pop] = 2 2
    - Queue: [push 2, push 1, pop, push 2, push 3, pop] = 2 3

Tổng kết

- Phân biệt:
  - Mảng
  - Danh sách liên kết đơn
  - Danh sách liên kết kép
- Phân biệt:
  - Ngăn xếp (Stack)
  - Hàng đợi (Queue)
- Chú ý các trường hợp của input
- Kiểm tra bài thực hành tuần trước ngày mai, yêu cầu các nhóm đi học đầy đủ
- Bài thực hành của lý thuyết hôm nay sẽ được cập nhật và cuối tuần