Chuong 1: Cơ bản về tổ hợp & Mệnh đề Logic

- Các nguyên lý
- Giải tích tổ hợp
- Hoán vị lặp, tổ hợp lặp
- Mệnh đề
- Bài toán tối ưu
- Giải thuật ngăn xếp và hàng đợi
- Khái niệm thuật toán và độ phức tạp của thuật toán
- Cấu trúc dữ liệu kiểu tập họp và đếm các phần tử trong tập họp dữ liệu

➢ Các nguyên lý

❖ Nguyên lý cộng

Giả sử để làm công việc A có 2 phương pháp

- Phương pháp 1 có n cách làm
- Phương pháp 2 có m cách làm

Khi đó số cách làm công việc A là n+m

Ví dụ. An có 3 áo tay dài, 5 áo tay ngắn. Để chọn 1 cái áo thì An có mấy cách

➢ Các nguyên lý

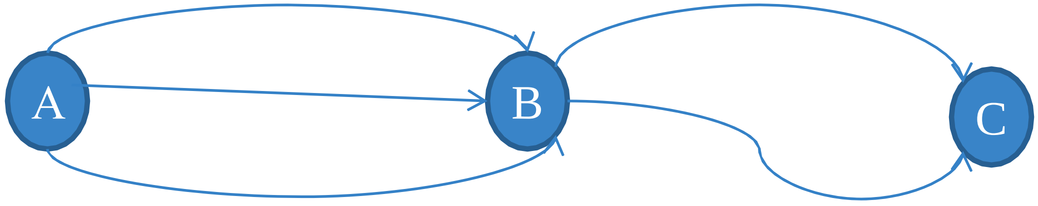
❖ Nguyên lý nhân

Giả sử để làm công việc A cần thực hiện 2 bước

- Bước 1 có n cách làm
- Bước 2 có m cách làm

Khi đó số cách làm công việc A là n.m

Ví dụ:



Có 3.2 =6 con đường đi từ A đến C

➢ Các nguyên lý

Ví dụ: Cho tập X ={1,2,3,4,5,0}

Hỏi có bao nhiêu số tự nhiên có 3 chữ số khác nhau mà chia hết cho 2

Giải. Gọi số có 3 chữ số là $\overline{abc}$

TH1 . c=0. Khi đó

c có 1 cách chọn

a có 5 cách chọn (a $\in$ X\{0})

b có 4 cách chọn (b $\in$ X\{a, 0})

TH1 có 1.4.5 =20

TH2 . c≠0. Khi đó

c có 2 cách chọn

a có 4 cách chọn (a $\in$ X\{c, 0})

b có 4 cách chọn (b $\in$ X\{a, c})

TH2 có 2.4.4 =32

Vậy có 20+32 =52

➢ Các nguyên lý

❖ **Nguyên lý chuồng bồ câu** (Derichlet)

Gọi $\lceil x \rceil$ là số nguyên nhỏ nhất lớn hơn hay bằng x.

Giả sử có n chim bồ câu ở trong k chuồng. Khi đó tồn tại ít nhất một chuồng chứa từ $\lceil n/k \rceil$ bồ câu trở lên.

Ví dụ. Có 20 chim bồ câu ở trong 7 cái chuồng. Khi đó sẽ có ít nhất 1 chuồng có 3 con bồ câu trở lên

- Trong 1 nhóm có 367 người thì ít nhất có 2 người sinh cùng ngày

➢ Các nguyên lý

Ví dụ. Cho tập X ={1,2,3,4,5,6,7,8,9}. Lấy A là tập hợp con của X gồm 6 phần tử. Khi đó trong A sẽ có hai phần tử có tổng bằng 10.

Giải.

Ta lập các chuồng như sau: {1,9} {2,8} {3,7} {4,6} {5}

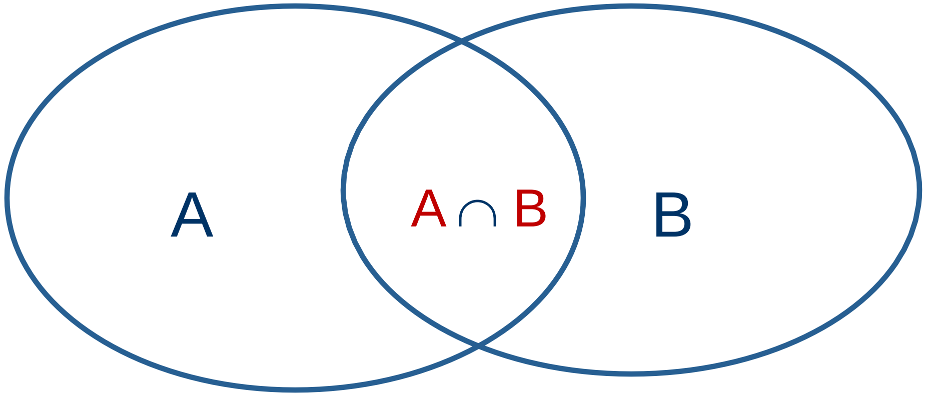
Do A có 6 phần tử nên trong 6 phần tử đó sẽ có 2 phần tử trong 1 chuồng. Suy ra đpcm

Các nguyên lý

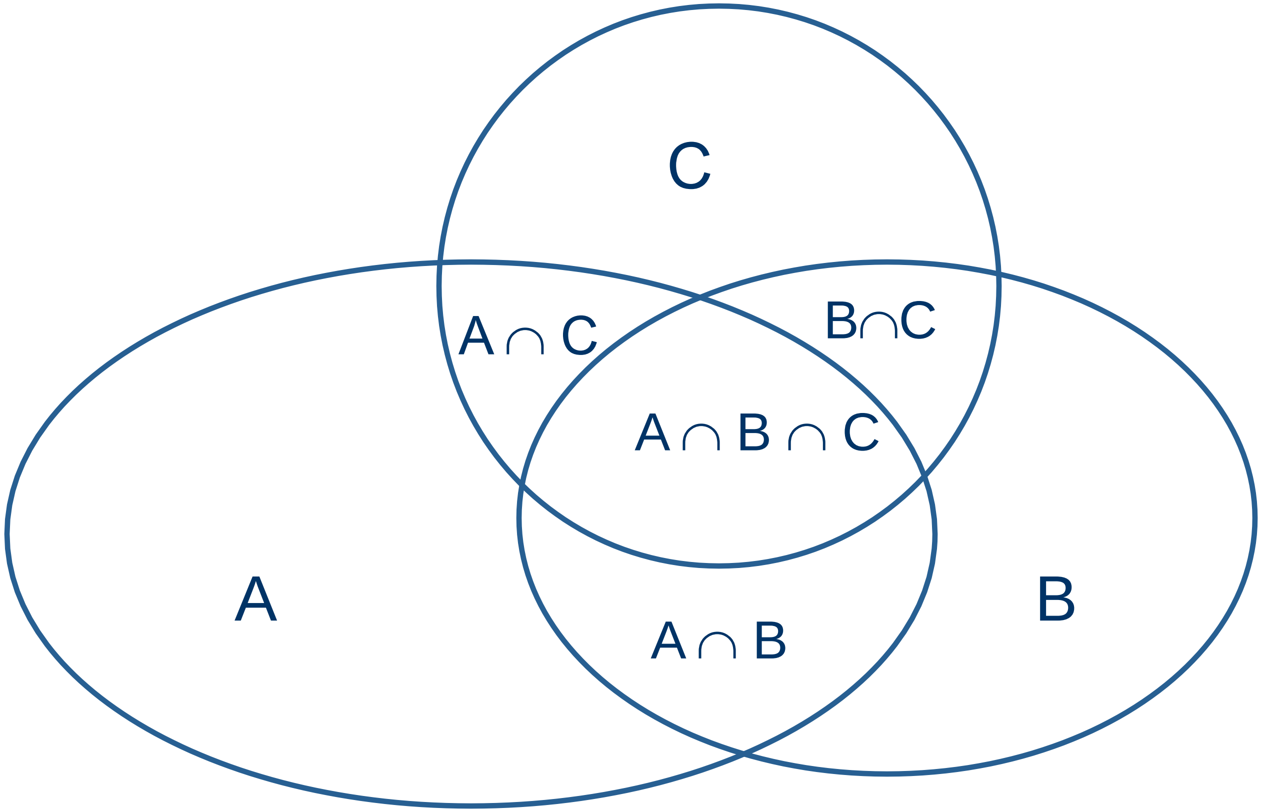
❖ Nguyên lý bù trừ.

Cho A và B là hai tập hữu hạn. Khi đó

$$|A \cup B| = |A| + |B| - |A \cap B|$$



➢ Các nguyên lý



$|A \cup B \cup C| = ?$

➢ Các nguyên lý

Ví dụ. Trong một lớp ngoại ngữ Anh Pháp. Có 24 HS học Tiếng Pháp, 26 học sinh học Tiếng Anh. 15 học sinh học Tiếng Anh và Tiếng Pháp. Hỏi lớp có bao nhiêu người

Giải.

Gọi A là những học sinh học Tiếng Pháp

B là những học sinh học Tiếng Anh

Khi đó. Số học sinh của lớp là $|A \cup B|$. Theo nguyên lý bù trừ ta có $|A \cup B| = |A|+|B| - |A \cap B|=24+26-15=35$

Giải tích tổ hợp

1. Hoán vị

Định nghĩa. Cho tập hợp A gồm n phần tử. Mỗi cách sắp đặt có thứ tự n phần tử của A được gọi là một *hoán vị của n phần tử*. Số các hoán vị của n phần tử ký hiệu là P_n

$$P_n = n! = n.(n-1).(n-2)\ldots 1$$

Quy ước $0! = 1$

Ví dụ. Cho A ={a,b,c}. Khi đó A có các hoán vị sau

abc,acb,

bac,bca,

cab,cba

Giải tích tổ hợp

Ví dụ. Nếu A là tập hợp n phần tử thì số song ánh từ A vào A là n!

Cho X ={1,2,3,4,5}. Hỏi có bao nhiêu số tự nhiên gồm 5 chữ số khác nhau được tạo từ tập X ➔ 5!

➢ Giải tích tổ hợp

❖ Chỉnh hợp.

Định nghĩa. Cho A là tập hợp gồm n phần tử. Mỗi bộ gồm k phần tử ($1 \leq k \leq n$) sắp thứ tự của tập hợp A được gọi là một *chỉnh hợp chập k của n* phần tử.

Số các chỉnh hợp chập k của n ký hiệu là A_n^k

- Công thức

$$A_n^k = \frac{n!}{(n-k)!}$$

Ví dụ. Cho X ={abc}. Khi đó X có các chỉnh hợp chập 2 của 3 là: ab, ba, ac, ca, bc, cb.

Giải tích tổ hợp

Ví dụ. Có bao nhiêu số tự nhiên gồm 3 chữ số được tạo thành từ 1,2,3,4,5,6.

Kết quả: A_6^3

➢ Giải tích tổ hợp

❖ Tổ hợp.

Định nghĩa. Cho tập hợp A gồm n phần tử. Mỗi tập con gồm k phần tử của A được gọi là một *tổ hợp chập k của n phần tử.*

Số *tổ hợp chập k của n phần tử được kí hiệu là* C_n^k *hay* $\begin{pmatrix} n \\ k \end{pmatrix}$

$$C_n^k = \frac{n!}{k!(n-k)!}$$

Tính chất $\quad C_n^{n-k} = C_n^k \qquad C_n^k + C_n^{k-1} = C_{n+1}^k$

Giải tích tổ hợp

Ví dụ. Cho X = {1,2,3,4}. Tổ hợp chập 3 của 4 phần tử của X là {1,2,3}, {1,2,4}, {1,3,4} , {2,3,4}

Một lớp có 30 học sinh. Hỏi có bao nhiêu cách chọn 10 bạn

- Số cách chọn là tổ hợp chập 10 của 30. C_{30}^{10}

➢ Hoán vị lặp, tổ hợp lặp

1. Hoán vị lặp

Định nghĩa. Cho n đối tượng trong đó có n_i đối tượng loại i giống hệt nhau (i =1,2,…,k ; n_1+ n_2,…+ n_k= n).

Mỗi cách sắp xếp có thứ tự n đối tượng đã cho gọi là một *hoán vị lặp* của n.

Số hoán vị của n đối tượng, trong đó có n_1 đối tượng giống nhau thuộc loại 1, n_2 đối tượng giống nhau thuộc loại 2,…, n_k đối tượng giống nhau thuộc loại k, là

$$\frac{n!}{n_1!n_2!\ldots n_k!}$$

➢ Hoán vị lặp, tổ hợp lặp

Ví dụ. Có bao nhiêu chuỗi kí tự khác nhau bằng cách sắp xếp các chữ cái của từ SUCCESS?

Giải. Trong từ SUCCESS có 3 chữ S, 1 chữ U, 2 chữ C và 1 chữ E. Do đó số chuỗi có được là

.

$$\frac{7!}{3!1!2!1!} = 420$$

➢ Hoán vị lặp, tổ hợp lặp

❖ Tổ hợp lặp

Định nghĩa. Mỗi cách chọn ra k vật từ n loại vật khác nhau (trong đó mỗi loại vật có thể được chọn lại nhiều lần) được gọi là *tổ hợp lặp chập k của n*

Số các tổ hợp lặp chập k của n được ký hiệu là K_n^k

$$K_n^k = C_{n+k-1}^k$$

➢ Hoán vị lặp, tổ hợp lặp

Ví dụ. Có 3 loại nón A, B, C. An mua 2 cái nón. Hỏi An có bao nhiêu cách chọn.

Ta có mỗi cách chọn là mỗi tổ hợp lặp chập 2 của 3. Cụ thể AA, AB, AC, BB, BC, CC

$$K_3^2 = C_{3+2-1}^2 = C_4^2 = 6$$

➢ Hoán vị lặp, tổ hợp lặp

Hệ quả. Số nghiệm nguyên không âm ($x_1,x_2,\ldots,x_n$) (mỗi x_i đều nguyên không âm) của phương trình

$$x_1 + x_2 + \ldots + x_n = k \text{ là}$$

$$K_n^k = C_{n+k-1}^k$$

Số cách chia k vật đồng chất nhau vào n hộp phân biệt cũng chính bằng số tổ hợp lặp chập k của n

$$K_n^k = C_{n+k-1}^k$$

➢ Hoán vị lặp, tổ hợp lặp

Ví dụ. Tìm số nghiệm nguyên không âm của phương trình

$x_1 + x_2 + x_3 + x_4 = 20$ (1)

Thỏa điều kiện $x_1 \le 3; x_2 \ge 2; x_3 > 4$ (*).

Giải. Ta viết điều kiện đã cho thành $x_1 \le 3; x_2 \ge 2; x_3 \ge 5.$

Xét các điều kiện sau:

$x_2 \ge 2; x_3 \ge 5$ (**)

$x_1 \ge 4; x_2 \ge 2; x_3 \ge 5$ (***)

Gọi p, q, r lần lượt là các số nghiệm nguyên không âm của phương trình (1) thỏa các điều kiện (*), (**), (***). Ta có:

➢ Hoán vị lặp, tổ hợp lặp

$p = q – r$.

Trước hết ta tìm q.

Đặt

$x_1' = x_1$; $x_2' = x_2 – 2$; $x_3' = x_3 - 5$; $x_4' = x_4$

Phương trình (1) trở thành

$x_1' + x_2' + x_3' + x_4' = 13$ (2)

Số nghiệm nguyên không âm của phương trình (1) thỏa điều kiện (**) bằng số nghiệm nguyên không âm của phương trình (2)

➢ Hoán vị lặp, tổ hợp lặp

Số nghiệm đó là $K_4^{13} = C_{4+13-1}^{13} = C_{16}^{13}$.

Vậy $q = C_{16}^{13}$.

Lý luận tương tự, ta có $r = K_4^9 = C_{4+9-1}^9 = C_{12}^9$

$$p = q - r = C_{16}^{13} - C_{12}^{9} = 560 - 220 = 340.$$

Suy ra. Vậy số nghiệm nguyên không âm của phương trình (1) thỏa điều kiện $(*)$ là 340

Hoán vị lặp, tổ hợp lặp

Số nghiệm đó là $K_4^{13} = C_{4+13-1}^{13} = C_{16}^{13}$.

Vậy $q = C_{16}^{13}$.

Lý luận tương tự, ta có $r = K_4^{9} = C_{4+9-1}^{9} = C_{12}^{9}$

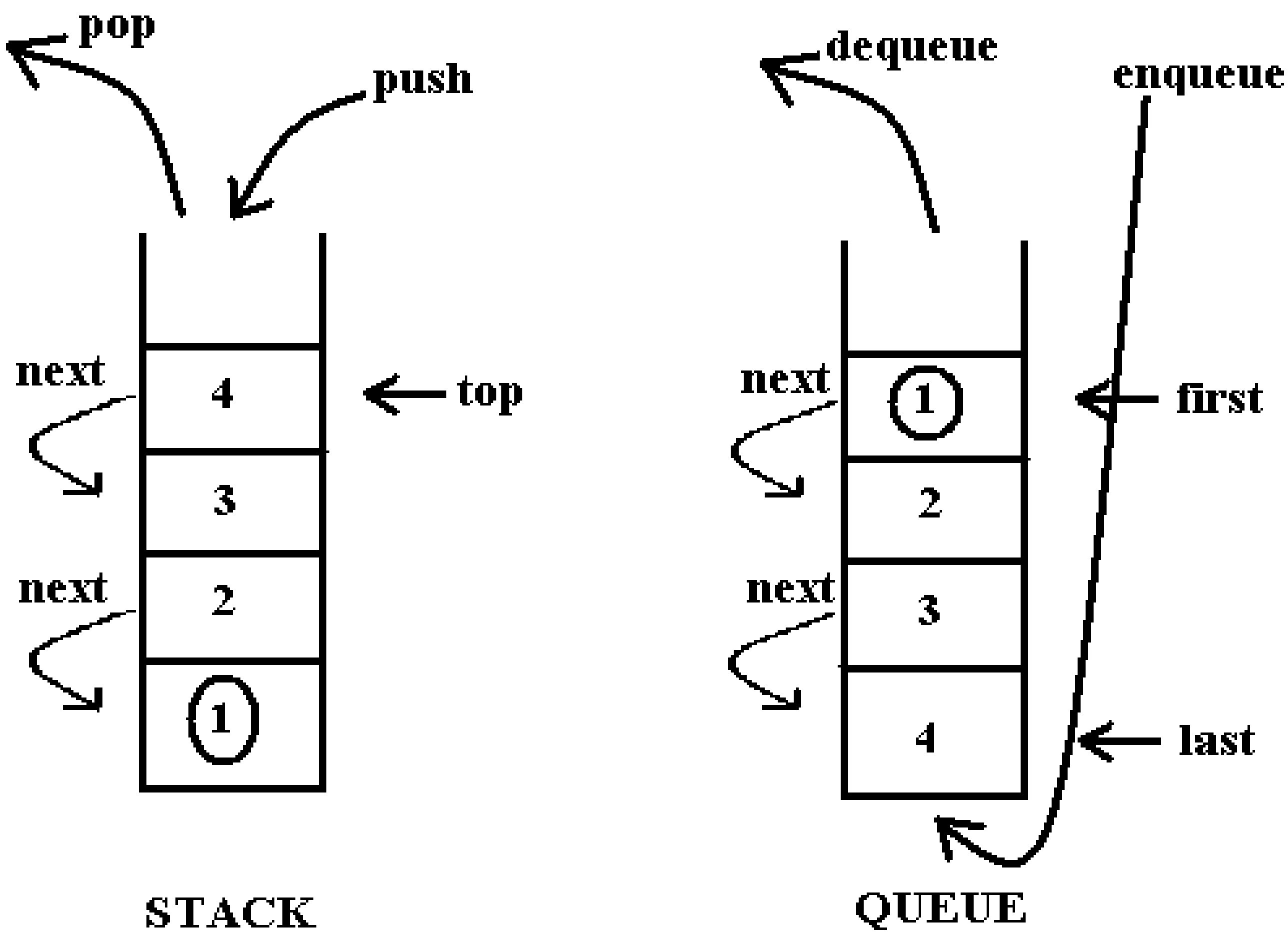
$$p = q - r = C_{16}^{13} - C_{12}^{9} = 560 - 220 = 340.$$

Suy ra. Vậy số nghiệm nguyên không âm của phương trình (1) thỏa điều kiện $(*)$ là 340

➢ Ngăn xếp & hàng đợi (Stack & Queue)

- ❖ Ngăn xếp (Stack) và Hàng đợi (Queue) là hai trong số những cấu trúc dữ liệu cực kỳ quan trọng, được sử dụng thường xuyên trong thiết kế thuật toán.
- ❖ Chính máy tính cũng sử dụng nhiều ứng dụng của ngăn xếp (chẳng hạn như việc quản lý bộ nhớ trong khi thi hành chương trình, hay lưu trữ các lời gọi đệ quy,...).
- ❖ Về bản chất, ngăn xếp và hàng đợi cũng giống như mảng, chúng là một tập hợp các phần tử cùng kiểu dữ liệu, nhưng được lưu trữ có tính thứ tự.

Ngăn xếp & hàng đợi (Stack & Queue)



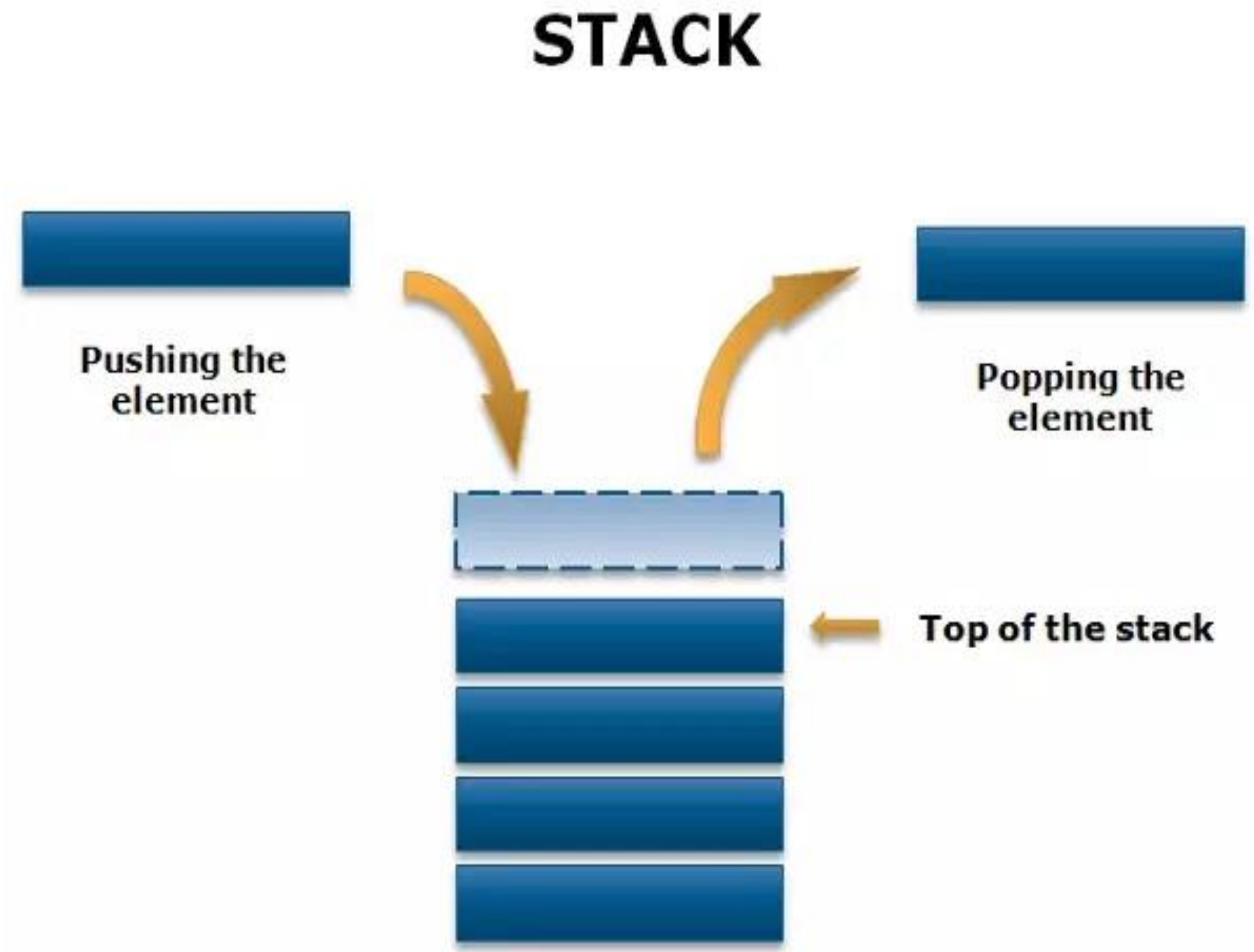
LIFO (Last In First Out) ***FIFO (First In First Out)***

➢ Ngăn xếp & hàng đợi (Stack & Queue)

- Ngăn xếp (Stack): Ngăn xếp là một kiểu danh sách mà việc bổ sung một phần tử và xóa một phần tử được thực hiện ở cuối danh sách.
- Ngăn xếp giống như một chồng đĩa. Muốn thêm một chiếc đĩa vào thì phải đặt nó lên đỉnh của chồng đĩa (phía cuối), và muốn lấy một chiếc đĩa ra thì cũng phải lấy từ trên xuống.

➢ Ngăn xếp & hàng đợi (Stack & Queue)

❖ Ngăn xếp (Stack):



➢ Ngăn xếp & hàng đợi (Stack & Queue)

❖ Ngăn xếp (Stack):

Phần tử ở đỉnh ngăn xếp (cuối danh sách) được gọi là phần tử ***top*** của ngăn xếp. Nguyên tắc thêm - xóa phần tử như trên được gọi là "vào sau ra trước", do đó ngăn xếp còn có tên gọi khác là ***danh sách kiểu LIFO (Last In First Out).***

➢ Ngăn xếp & hàng đợi (Stack & Queue)

❖ Ngăn xếp (Stack):

Có 6 thao tác cơ bản ngăn xếp cung cấp:

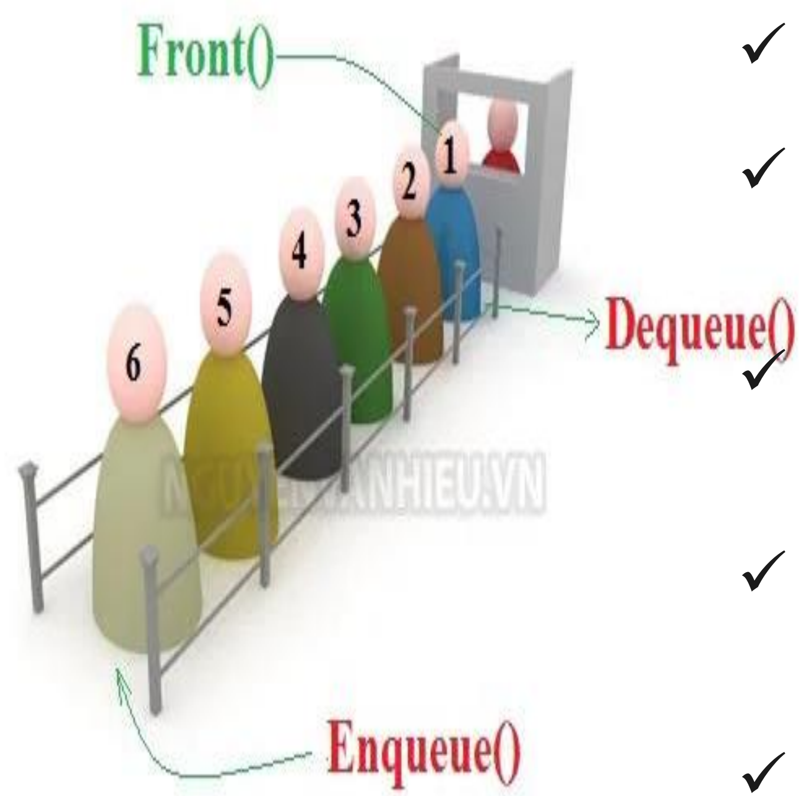
- ✓ initinit: Khởi tạo ngăn xếp rỗng.
- ✓ is_emptyis_empty: Kiểm tra xem ngăn xếp có rỗng không.
- ✓ is_fullis_full: Kiểm tra xem ngăn xếp có bị đầy (tràn) hay không.
- ✓ get_topget_top: Trả về phần tử ở đỉnh ngăn xếp.
- ✓ pushpush: Thêm một phần tử vào ngăn xếp.
- ✓ poppop: Lấy ra phần tử ở đỉnh ngăn xếp.

➢ Ngăn xếp & hàng đợi (Stack & Queue)

- **Hàng đợi (Stack)**: Hàng đợi là một cấu trúc dữ liệu biểu diễn một danh sách các phần tử đứng trong "hàng chờ" được xử lý. Trong cấu trúc dữ liệu này, việc bổ sung một phần tử được thực hiện ở cuối danh sách, còn việc loại bỏ một phần tử được thực hiện ở đầu danh sách.
- Vì nguyên tắc "vào trước ra trước" như vậy nên hàng đợi còn được gọi là danh sách kiểu **FIFO (First In First Out)**.
- Phần tử ở đầu hàng đợi sẽ gọi là phần tử front, còn phần tử ở cuối hàng đợi gọi là phần tử rear.

➢ Ngăn xếp & hàng đợi (Stack & Queue)

❖ Hàng đợi (Queue)



Tương tự như ngăn xếp, có 6 thao tác cơ bản trên hàng đợi:

- ✓ initinit: Khởi tạo một hàng đợi rỗng.
- ✓ is_emptyis_empty: Kiểm tra hàng đợi có rỗng hay không.
- ✓ is_fullis_full: Kiểm tra hàng đợi đã bị đầy chưa.
- ✓ get_frontget_front: Trả về giá trị của phần tử ở đầu hàng đợi.
- ✓ pushpush: Đẩy một phần tử vào cuối hàng đợi.
- ✓ poppop: Loại bỏ một phần tử ở đầu hàng đợi.

➢ Khái niệm Thuật toán - Algorithm

- ❖ Thuật toán là một phương thức gồm tập hợp các câu lệnh hay các bước được thực hiện theo một thứ tự nhất định nhằm giải quyết một vấn đề logic nào đó.
- ❖ Từ thuật toán còn được gọi là Algorithm, nó được đặt theo tên của một nhà toán học người Trung Á là al'Khwarizmi. Ông là tác giả của một cuốn sách về số học, trong đó ông đã dùng phương pháp mô tả rất rõ ràng và mạch lạc cách giải những bài toán. Sau này, phương pháp này đã được xem là một chuẩn mực và được nhiều nhà toán học khác áp dụng theo.

Khái niệm Thuật toán - Algorithm

- Thuật toán máy tính:

Trong toán học và khoa học máy tính, thuật toán máy tính, hay còn gọi là giải thuật, là một tập hợp các hướng dẫn được xác định rõ ràng, có thể thực hiện được bằng máy tính, nhằm giải quyết một lớp vấn đề hoặc để thực hiện một phép tính.

- Các thiết bị máy tính sử dụng thuật toán để thực hiện các chức năng của nó một cách rõ ràng như thực hiện các phép tính, suy luận tự động, xử lý dữ liệu (database), và các tác vụ khác.

Thuật toán và cấu trúc dữ liệu

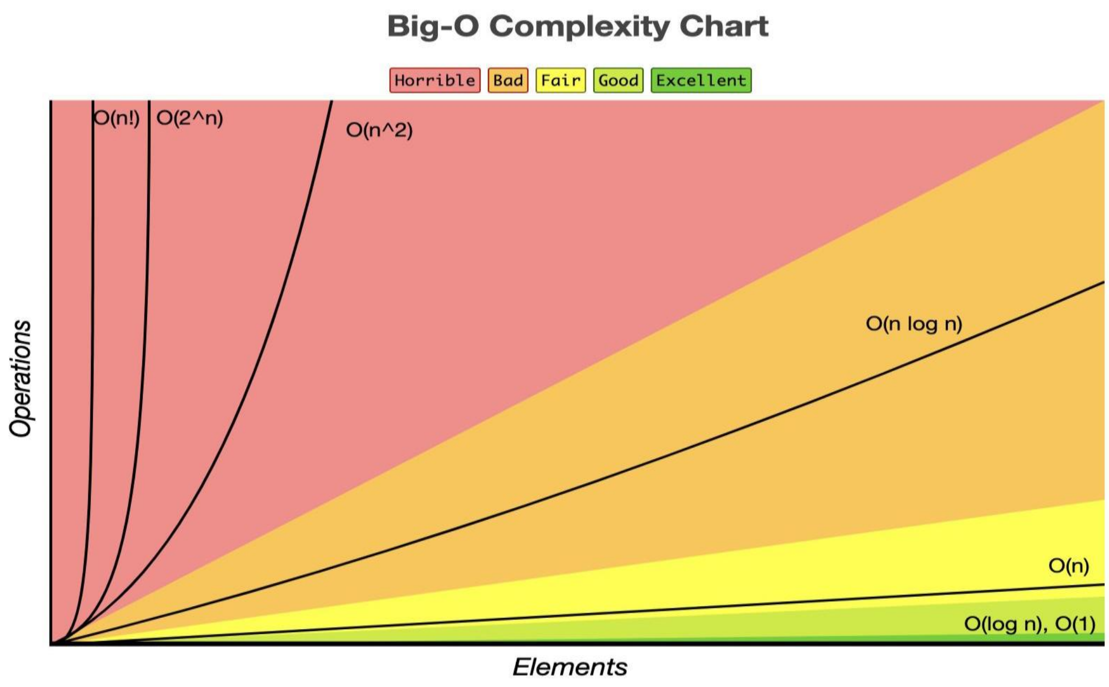
- Thuật toán là một tập hợp các bước để giải quyết một vấn đề cụ thể.
- Còn cấu trúc dữ liệu là một vị trí được đặt tên và có thể được sử dụng để lưu trữ, tổ chức dữ liệu.
- Thuật toán và cấu trúc dữ liệu cho phép chúng ta viết các chương trình trên laptop một cách hiệu quả và tối ưu. Hai thành phần này hỗ trợ nhau, các thuật toán sẽ cần cấu trúc dữ liệu bên trong để làm cho chúng hoạt động theo đúng kỳ vọng.
- Ví dụ, nếu chúng ta cần sắp xếp một danh sách thì có thể sử dụng **cấu trúc dữ liệu mảng để lưu trữ và dùng thuật toán để sắp xếp** các phần tử trong danh sách đó theo mong muốn.

Độ phức tạp thuật toán

- Độ phức tạp của thuật toán (biểu diễn bằng **Big-O Notation**), là biểu thức mô tả hành vi thuật toán (ví dụ, về mặt thời gian tính toán hoặc lượng bộ nhớ cần dùng) khi kích thước dữ liệu thay đổi.
- Big O mô tả mối liên hệ giữa số lượng phần tử đầu vào và số lượng operation – thời gian chạy, hoặc số lượng bộ nhớ mà thuật toán cần sử dụng.
- Với các thuật toán có độ phức tạp là O(n), khi dữ liệu đầu vào có 10 phần tử, chương trình phải chạy 10 operation. Khi số dữ liệu đầu vào tăng gấp đôi, số lượng câu lệnh phải tăng gấp đôi
- Nếu độ phức tạp là **$O(n^2)$**, khi số lượng dữ liệu đầu vào tăng gấp đôi, số lượng câu lệnh phải tăng gấp **2^2** tức là gấp 4 lần

➢ Độ phức tạp thuật toán

- ❖ Thuật toán có độ phức tạp càng lớn, khi số lượng dữ liệu càng nhiều hơn lên thì nó sẽ chạy càng chậm.
- ❖ Ví dụ, với bài toán tìm đường cho người đưa thư qua nhiều thành phố (traveling salesman problem), dùng thuật toán vét cạn có độ phức tạp là O(n!), khi dữ liệu quá nhiều tầm vài ngàn địa điểm thì máy tính cũng chạy rất lâu.



➢ Xác định độ phức tạp của thuật toán

Cách tính độ phức tạp của một số giải thuật đơn giản.

❖ QUY TẮC XÁC ĐỊNH ĐỘ PHỨC TẠP

- Độ phức tạp tính toán của giải thuật: O(f(n))
- Việc xác định độ phức tạp tính toán của giải thuật trong thực tế có thể tính bằng một số quy tắc đơn giản sau:
  - **Quy tắc bỏ hằng số**:

    T(n) = O(c.f(n)) = O(f(n)) với c là một hằng số dương
  - **Quy tắc lấy max**:

    T(n) = O(f(n)+ g(n)) = O(max(f(n), g(n)))
  - **Quy tắc cộng**:

    T1(n) = O(f(n)) T2(n) = O(g(n))

    T1(n) + T2(n) = O(f(n) + g(n))
  - **Quy tắc nhân**:

    Đoạn chương trình có thời gian thực hiện T(n)=O(f(n))

    Nếu thực hiện k(n) lần đoạn chương trình với k(n) = O(g(n)) thì độ phức tạp sẽ là O(g(n).f(n))

➢ Xác định độ phức tạp của thuật toán

PHÉP TOÁN TÍCH CỰC (BEST PROXY)

❖ Trong một thuật toán, ta chú ý đặc biệt đến một phép toán gọi là phép toán tích cực. Đó là phép toán mà số lần thực hiện không ít hơn các phép toán khác

- Ví dụ 1:

```
s = 0;
for (i=0; i<=n;i++){
        p =1;
        for (j=1;j<=i;j++)
                p=p * x / j;
        s = s+p;
}
```

Phép toán tích cực là p = p * x / j

Số lần thực hiện phép toán này là

$$0+1+2+\ldots+n = n(n-1)/2 = n^2/2 - n/2$$

=> Vậy độ phức tạp là $O(n^2/2 - n/2)$

$= O(n^2/2)$ sử dụng quy tắc lấy max

$= O(n^2)$ sử dụng quy tắc bỏ hằng số

➢ Xác định độ phức tạp của thuật toán

❖ <u>Ví dụ 2:</u>

```
s = 1; p = 1;
for (i=1;i<=n;i++) {
        p = p * x / i;
        s = s + p;
}
```

Phép toán tích cực là p = p * x / i

Số lần thực hiện phép toán là n

=> Vậy độ phức tạp của thuật toán là O(n)

❖ <u>Ví dụ 3:</u>

```
s=n*(n-1) /2;
```

=> Độ phức tạp của thuật toán là O(1), nghĩa là thời gian tính toán không phụ thuộc vào n

➢ Xác định độ phức tạp của thuật toán

❖ Ví dụ 4:

Thuật toán tính tổng các số từ 1 đến n

```
s=0;
for (i= 1;i<=n;i++)
        s=s+i;
```

Vòng lặp lặp n lần phép gán s = s+i, nên thời gian tính toán tỉ lệ thuận với n, tức độ phức tạp là O(n).

=> độ phức tạp là O(n)

❖ Ví dụ 5:

```
for (i= 1;i<=n;i++)
        for (j= 1;j<=n;j++)
                //Lệnh
```

=> Dùng quy tắc nhân ta có O(n^2)

tương tự như vậy ta sẽ có O(n^3), O(n^4).

➢ Xác định độ phức tạp của thuật toán

❖ Ví dụ 6:

```
for (i= 1;i<=n;i++)
        for (j= 1;j<=m;j++)
                //Lệnh
```

=> Dùng quy tắc nhân ta có O(n*m)

❖ Ví dụ 7:

```
for (i= 1;i<=n;i++)
        //lệnh1
for (j= 1;j<=m;j++)
        //lệnh 2
```

=> Dùng quy tắc cộng và quy tắc lấy max, sẽ có O(max (n,m))

➢ Xác định độ phức tạp của thuật toán

❖ Ví dụ 8:

```
for (i= 1;i<=n;i++) {
        for (u= 1;u<=m;u++)
                for (v= 1;v<=n;v++)
                        //lệnh
        for j:= 1 to x do
                for k:= 1 to z do
                        //lệnh
}
```

=> O(n*max (n*m,x*z))

➢ Xác định độ phức tạp của thuật toán

❖ Ví dụ 9:

```
for (i= 1;i<=n;i++)
        for (j= 1;j<=m;j++) {
                for (k= 1;k<=x;k++)
                        //lệnh
                for (h= 1;h<=y;h++)
                        //lệnh
        }
```

=> O(n*m* max (x,y))

Xác định độ phức tạp của thuật toán

❖ <u>Ví dụ 10:</u>

```
for (i= 1;i<=n;i++)
        for (u= 1;u<= m;u++)
                for (v= 1;v<=n;v++)
                        //lệnh
for (j= 1;j<=x;j++)
                for (k= 1;k<=z;k++)
                        //lệnh
=> O(max (m*n^2,x*z))
```

➢ Xác định độ phức tạp của thuật toán

❖ Ví dụ 11: Đoạn chương trình tính tổng 2 đa thức

$P(x) = x_m x^m + a_{m-1} x^{m-1} + \ldots + a_1 x + a_0$

$Q(x) = b_n x^n + b_{n-1} x^{n-1} + \ldots + b_1 x + b_0$

```
if (m<n) p = m; else p =n;
for (i=0;i<=p;i++)
        c[i]=a[i] + b[i];
if (p<m)
        for (i=p+1;i<=m;i++) c[i] = a[i];
else
        for (i=p+1;i<=n;i++) c[i] = b[i];
while (p>0 && c[p] = 0) p = p-1;
```

=> Độ phức tạp: O(max(m,n)

➢ Xác định độ phức tạp của thuật toán

❖ <u>Ví dụ 12:</u> Đoạn chương trình tính tích hai đa thức

$P(x) = x_m x^m + a_{m-1} x^{m-1} + \ldots + a_1 x + a_0$

$Q(x) = b_n x^n + b_{n-1} x^{n-1} + \ldots + b_1 x + b_0$

```
p = m+n;
for (i=0;i<=p;i++) c[i] = 0;
for (i=0;i<=m;i++)
        for (j=0;j<=n;j++)
                c[i+j] = c[i+j] + a[i] + b[j];
```

=> Độ phức tạp là O(m.n)

- **Sắp xếp theo chiều tăng của độ phức tạp, các độ phức tạp đặt trên cùng hàng là tương đương**

$1, 2^{100}$

$\log n$

$n \log n, \log (n!)$

n^2

$(\log n)^{\log n}, n^{\log(\log n)}$

2^n

3^n

$n!$

➢ Cấu trúc dữ liệu kiểu tập hợp và đếm các phần tử trong tập hợp dữ liệu

- ❖ SET (tập hợp): Cấu trúc dữ liệu đơn giản nhất đó là **mảng (array).** Ngoài cấu trúc dữ liệu: **queue và stack**.
- ❖ Set cũng là 1 cấu trúc dữ liệu để lưu một danh sách các phần tử cùng kiểu dữ liệu, trong đó các phần tử **không được trùng nhau**. Ví dụ như **array có thể có các phần tử trùng nhau như {1, 2, 1, 1, 2} nhưng set thì chỉ chứa các phần tử khác nhau {1, 2}**
- ❖ Khi thực hiện, set lại chỉ chứa các phần tử khác nhau, là vì khi insert mỗi phần tử x vào trong set, set sẽ kiểm tra và chỉ thêm phần tử đó khi x **không có** trong set.
- ❖ Khi chèn/ thêm mới 1 phần tử vào trong set, thì set sẽ kiểm tra xem có phần tử trong tập hợp chưa? Bài toán trong việc cài đặt set quy về việc làm sao trả lời câu hỏi phần tử có trong tập hợp ban đầu hay chưa?

Cấu trúc dữ liệu kiểu tập hợp và đếm các phần tử trong tập hợp dữ liệu

- Bài toán đặt ra là nếu có 1 dãy các phần tử a1, a2, ... an, hãy trả lời câu hỏi có tồn tại x trong dãy này không. Thông thường nếu các bạn duyệt từ 1 tới n để kiểm tra, độ phức tạp sẽ là O(n).
- Tuy nhiên nếu dãy a1->an là dãy sắp xếp tăng dần, thì việc tìm kiếm x có trong dãy hay không bằng phương pháp tìm kiếm nhị phân sẽ có độ phức tạp là **o(logn)**.
- Đối với C++, có 2 lớp cài đặt/biểu diễn set hay được dùng là **set và unordered_set**. Trong java thì có 3 cách cài đặt set là **HashSet, TreeSet và LinkedHashSet**.
- Set của C++ và treeset của Java thì cài đặt dựa trên tree, và dùng tư tưởng tìm kiếm nhị phân, trong khi unordered_set của C++ và hashset của java thì cài đặt dựa trên bảng băm. Mỗi lớp sẽ có sự khác nhau về hiệu quả trong từng trường hợp.

➢ Bài toán ứng dụng trong đếm cấu trúc dữ liệu

❖Do set chỉ chứa các phần tử không trùng nhau, nên có thể sử dụng cấu trúc dữ liệu này cho 1 số bài toán liên quan tới tập hợp như các bài toán sau:

Bài toán 1: Cho dãy n phần tử. Hãy đếm số phần tử khác nhau trong mảng bài này nếu không dùng set, bạn có thể đánh dấu mỗi phần tử trong mảng có hay chưa với dữ liệu nhỏ. Nhưng nếu giá trị của các phần tử rất lớn (Ví dụ từ 1 tới 10^9, thì lưu 1 mảng chứa các phần tử không trùng nhau. Mỗi khi thêm phần tử mới thì kiểm tra trùng hay không và thêm vào mảng. Đoạn mã sau đây demo bằng C++:

```
set<int> s;
for (int i=0; i<n; i++)
insert(a[i]);
return s.size();
```

➢ Bài toán ứng dụng trong đếm cấu trúc dữ liệu

❖ **Bài toán 2**: Cho danh sách chứa thời gian quẹt thẻ của nhân viên trong ngày. Hãy kiểm tra xem có bao nhiêu người khác nhau đi làm trước 9h sáng. Dùng tên nhân viên nếu tên nhân viên không trùng nhau (hoặc mã nhân viên) để đưa vào tập hợp. Đoạn code trong C++.

```
set<string> s;
for (int i=0; i<n; i++)
  if (a[i].time.hours <= 9) s.insert(a[i].name);
return s.size();
```

CƠ SỞ LOGIC

Nội dung

- Logic mệnh đề
- Các phương pháp chứng minh
- Suy diễn logic
- Vị từ và lượng từ

Logic mệnh đề

➢ Một số khái niệm của logic mệnh đề

- **Mệnh đề**: là một câu khẳng định hoặc đúng hoặc sai.
- Ví dụ:
  - Mặt trời quay quanh trái đất.
  - 1+1 = 2.
  - Hôm nay trời đẹp quá! (không là mệnh đề)
  - Học bài đi! (không là mệnh đề)
  - 3 là số chẵn phải không? (không là mệnh đề)
- Giá trị của một mệnh đề đúng được ký hiệu là T, giá trị mệnh đề sai được ký hiệu là F. Tập giá trị T, F còn được gọi là giá trị chân lý của một mệnh đề.

Bài tập làm nhanh

Kiểm tra các khẳng định sau có phải là mệnh đề không?

- Paris là thành phố của Mỹ.
- n là số tự nhiên.
- con nhà ai mà xinh thế!
- 3 là số nguyên tố.
- Bạn có khỏe không?
- $x^2 + 1$ luôn dương.

➢ Mệnh đề

❑ **Phân loại:** gồm 2 loại

a. **Mệnh đề sơ cấp** (nguyên thủy): thường là một mệnh đề khẳng định đơn.

b. **Mệnh đề phức hợp:** là mệnh đề được xây dựng từ các mệnh đề sơ cấp nhờ liên kết bằng các liên từ (**và, hay, khi và chỉ khi**,…) hoặc trạng từ "**không**".

<u>Ví dụ</u>:

- 2 **không** là số nguyên tố
- 2 là số nguyên tố (sơ cấp)
- **Nếu** trời đẹp **thì** tôi đi dạo
- An đang xem phim **hay** An đang học bài
- Lúc này trời đang mưa **và** trận đấu đang bị tạm hoãn

➢ Các phép toán mệnh đề

a. Phép phủ định: phủ định của mệnh đề P được ký hiệu là $\neg$P hay $\overline{P}$ ("không" P hay "phủ định của" P).

Bảng chân trị :

P	$\neg$P
1	0
0	1

Mệnh đề cho giá trị đúng nếu *p* sai và cho giá trị sai nếu *p* đúng.

Ví dụ :

+ 2 là số nguyên tố

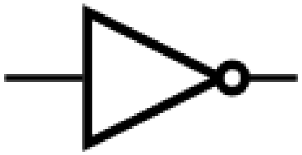
Phủ định: 2 **không** là số nguyên tố

+ $1 > 2$

Phủ định : $1 \leq 2$

➢ Các phép toán mệnh đề (cont)

a. Phép phủ định: ứng dụng mệnh đề phủ định: cổng (mạch) điện logic NOT.



Bảng chân lý - NOT Gate	
Đầu vào A	Đầu ra Q
0	1
1	0

➢ Các phép toán mệnh đề (cont)

b. Phép nối liền (hội, giao): của hai mệnh đề P, Q được kí hiệu bởi $P \wedge Q$ (đọc là "P và Q"), là mệnh đề được định bởi : $P \wedge Q$ đúng khi và chỉ khi P và Q đồng thời đúng.

Bảng chân trị

P	*Q*	*P*∧*Q*
0	0	0
0	1	0
1	0	0
1	1	1

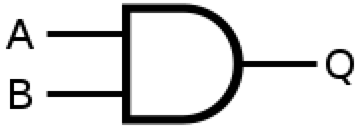
Mệnh đề có giá trị đúng khi cả *p* và *q* có giá trị đúng, có giá trị sai trong các trường hợp khác còn lại.

Ví dụ:

- 3 > 4 và Trần Hưng Đạo là một vị tướng
- 2 là số nguyên tố và 2 là số chẵn
- An đang hát và uống nước

➢ Các phép toán mệnh đề (cont)

b. <u>Phép nối liền</u> (hội, giao): ứng dụng phép hội (giao) trong mạch điện cổng logic AND



Truth Table - AND Gate		
Input A	Input B	Output Q
0	0	0
0	1	0
1	0	0
1	1	1

➢ Các phép toán mệnh đề (cont)

c. Phép nối rời (tuyển, hợp): của hai mệnh đề P, Q được kí hiệu bởi $P \vee Q$ (đọc là "P hay Q"), là mệnh đề được định bởi : $P \vee Q$ sai khi và chỉ khi P và Q đồng thời sai.

Bảng chân trị

P	*Q*	$P \vee Q$
0	0	0
0	1	1
1	0	1
1	1	1

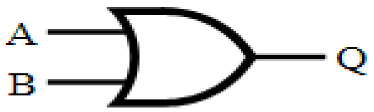
Mệnh đề có giá trị sai khi cả *p* và *q* có giá trị sai, có giá trị đúng trong các trường hợp khác còn lại.

Ví dụ:

- p > 4 hay p > 5
- 2 là số nguyên tố hay 2 là số chẵn

➢ Các phép toán mệnh đề (cont)

c. Phép nối rời (tuyển, hợp): ứng dụng của phép tuyển, hợp là phép logic OR, thể hiện qua mạch điện logic OR



Bảng chân lý - Cổng OR		
Đầu vào A	Đầu vào B	Đầu ra Q
0	0	0
0	1	1
1	0	1
1	1	1

➢ Các phép toán mệnh đề (cont)

d. Phép tuyển loại: của hai mệnh đề P, Q được kí hiệu bởi $P \oplus Q$ (đọc là "hoặc P hoặc Q"), là mệnh đề được định bởi : $P \oplus Q$ đúng khi một trong P hoặc Q có giá trị đúng.

Bảng chân trị

P	*Q*	$P \oplus Q$
0	0	0
0	1	1
1	0	1
1	1	0

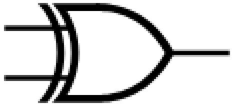
Mệnh đề có giá trị đúng khi một trong *p* hoặc *q* có giá trị đúng, có giá trị sai trong các trường hợp khác còn lại.

Ví dụ:

- $\pi > 4$ hay $\pi > 5$
- 2 là số nguyên tố hay 2 là số chẵn

➢ Các phép toán mệnh đề (cont)

d. Phép tuyển loại: ứng dụng của phép tuyển loại là mạch logic XOR trong mạch điện



Bảng chân lý - Cổng XOR		
Đầu vào A	Đầu vào B	Đầu ra Q
0	0	0
0	1	1
1	0	1
1	1	0

➢ Các phép toán mệnh đề (cont)

Ví dụ

- “Hôm nay, An giúp mẹ lau nhà và rửa chén”
- “Hôm nay, cô ấy đẹp và thông minh ”
- “Ba đang đọc báo hay xem phim”

➢ Các phép toán mệnh đề (cont)

e. Phép kéo theo: Mệnh đề P kéo theo Q của hai mệnh đề P và Q, ký hiệu bởi $P \rightarrow Q$ (đọc là "P kéo theo Q" hay "Nếu P thì Q" hay "P là điều kiện đủ của Q" hay "Q là điều kiện cần của P") là mệnh đề được định bởi:

$P \rightarrow Q$ sai khi và chỉ khi P đúng mà Q sai.

Bảng chân trị

P	*Q*	P→Q
0	0	1
0	1	1
1	0	**0**
1	1	1

Mệnh đề có giá trị sai khi *p* đúng và *q* sai, có giá trị đúng trong các trường hợp khác còn lại.

➢ Các phép toán mệnh đề (cont)

Ví dụ:

- Nếu 1 = 2 thì Lenin là người Việt Nam

- Nếu trái đất quay quanh mặt trời thì 1 + 3 = 5

- $\pi > 4$ kéo theo $5 > 6$

- $\pi < 4$ thì trời mưa

- Nếu 2 + 1 = 0 thì tôi là chủ tịch nước

➢ Các phép toán mệnh đề (cont)

f. Phép kéo theo hai chiều: Mệnh đề P kéo theo Q và ngược lại của hai mệnh đề P và Q, ký hiệu bởi P ↔ Q (đọc là "P nếu và chỉ nếu Q" hay "P khi và chỉ khi Q" hay "P là điều kiện cần và đủ của Q" hay "P tương đương với Q"), là mệnh đề xác định bởi:

P ↔ Q đúng khi và chỉ khi P và Q có cùng chân trị

Bảng chân trị

P	*Q*	P↔Q
0	0	1
0	1	0
1	0	0
1	1	1

Mệnh đề có giá trị đúng khi *p* và *q* có cùng giá trị chân lý, có giá trị sai trong các trường hợp khác còn lại.

➢ Các phép toán mệnh đề (cont)

Ví dụ:

- 2=4 khi và chỉ khi 2+1=0
- 6 chia hết cho 3 khi và chi khi 6 chia hết cho 2
- London là thành phố nước Anh nếu và chỉ nếu thành phố HCM là thủ đô của VN
- π>4 là điều kiện cần và đủ của 5>6

➢ Dạng mệnh đề

1. Định nghĩa: Dạng mệnh đề là một biểu thức được cấu tạo từ:

- Các hằng mệnh đề
- Các biến mệnh đề *p, q, r, ...,* tức là các biến lấy giá trị là các mệnh đề nào đó
- Các phép toán $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$ và dấu đóng mở ngoặc ().

Dạng mệnh đề được gọi là hằng đúng nếu nó luôn lấy giá trị **1**

Dạng mệnh đề gọi là hằng sai (hay mâu thuẫn) nếu nó luôn lấy giá trị 0.

Ví dụ:

$E(p,q) = \neg(\neg p \wedge q)$

$E(p,q,r) = (p \rightarrow q) \wedge \neg(q \wedge r)$

➢ Dạng mệnh đề (Cont)

Bảng chân trị của dạng mệnh đề E(p,q,r): là bảng ghi tất cả các trường hợp chân trị có thể xảy ra đối với dạng mệnh đề E theo chân trị của các biến mệnh đề p, q, r.

Nếu có n biến, bảng này sẽ có 2^n dòng, chưa kể dòng tiêu đề.

Ví dụ:

$E(p,q,r) = (p \vee q) \rightarrow r$. Ta có bảng chân trị sau:

➢ Dạng mệnh đề (Cont)

Mệnh đề **E(p,q,r) =(p $\vee$q) $\rightarrow$r** theo 3 biến p,q,r có bảng chân trị sau

p	q	r	p$\vee$q	(p $\vee$q) $\rightarrow$r
0	0	0	0	1
0	0	1	0	1
0	1	0	1	0
0	1	1	1	1
1	0	0	1	0
1	0	1	1	1
1	1	0	1	0
1	1	1	1	1

➢ Dạng mệnh đề (Cont)

Bài tập: Lập bảng chân trị của những dạng mệnh đề sau

$A(p,q) = \neg(p \wedge q) \wedge p$

$B(p,q,r) = p \wedge (q \vee r) \leftrightarrow \neg q$

➢ Độ ưu tiên các phép toán

1. Ngoặc ()
2. Phủ định
3. Và
4. Hay
5. Kéo theo →
6. Kéo theo hai chiều

- Ví dụ:
  - $\mathbf{p \vee q \rightarrow r}$ hiểu là $\mathbf{(p \vee q) \rightarrow r}$
  - $\mathbf{p \wedge (q \vee r) \leftrightarrow \neg q}$ hiểu là $(\mathbf{p \wedge (q \vee r)) \leftrightarrow (\neg q)}$

➢ Bài tập làm nhanh

- Lập bảng chân lý cho các mệnh đề sau:

$$\neg p \rightarrow (p \vee q)$$

$$(p \vee \neg q) \wedge (\neg p \vee q)$$

$$(p \rightarrow q) \vee (q \rightarrow p)$$

➢ Bảng giá trị chân lý của các phép toán mệnh đề

p	q	$\neg p$	$p \wedge q$	$p \vee q$	$p \oplus q$	$p \rightarrow q$	$p \leftrightarrow q$
T	T	F	T	T	F	T	T
T	F	F	F	T	T	F	F
F	T	T	F	T	T	T	F
F	F	T	F	F	F	T	T

Các phép toán cấp bít ứng dụng trong ngôn ngữ LT				
Giá trị của A	Giá trị của B	A **and** B	A **or** B	A **xor** B
A = 13 =1101	B = 8=1000	1000	1101	0101

Tương đương logic

➢ Nhắc lại về Dạng mệnh đề (mệnh đề phức hợp)

- Một mệnh đề phức hợp luôn đúng với bất kể các giá trị chân lý của các mệnh đề thành phần được gọi là *hằng đúng*.
- Một mệnh đề phức hợp luôn sai với mọi giá trị chân lý của các mệnh đề thành phần được gọi là *mâu thuẫn*.

Ví dụ về mệnh đề hằng đúng & mệnh đề mâu thuẫn			
p	$\overline{p}$	$\boxed{p \vee \overline{p}}$	$\boxed{p \wedge \overline{p}}$
T	F	T	F
F	T	T	F

➢ Hai dạng mệnh đề **tương đương logic**

- **Ký hiệu**: A≡B, hoặc A⇔B, hoặc A=B
  - – khi và chỉ khi các cột cho giá trị chân lý của chúng giống nhau.
- **Ví dụ:** 2 mệnh đề phức hợp sau là tương đương loggic

Bảng giá trị chân lý đối với $\overline{(p \vee q)}$ *và* $\overline{p} \wedge \overline{q}$						
p	q	p∨q	$\boxed{\overline{(p \vee q)}}$	$\overline{p}$	$\overline{q}$	$\boxed{\overline{p} \wedge \overline{q}}$
T	T	T	**F**	F	F	**F**
T	F	T	**F**	F	T	**F**
F	T	T	**F**	T	F	**F**
F	F	F	**T**	T	T	**T**

Định lý: *Hai dạng mệnh đề A và B tương đương logic với nhau khi và chỉ khi A↔B là hằng đúng.*

➢ **Chứng minh được sự tương đương logic**

- Bằng phương pháp bảng chân lý, dễ dàng chứng minh được sự tương đương logic của các dạng mệnh đề dưới đây:

$$p \rightarrow q \Leftrightarrow \overline{p} \vee q$$

$$p \leftrightarrow q \Leftrightarrow (p \rightarrow q) \wedge (q \rightarrow p)$$

$$\overline{\overline{p}} \Leftrightarrow p$$

➢ Các luật tương đương logic

1. Phủ định của phủ định

$$\neg \neg p \Leftrightarrow p$$

2. Luật De Morgan

$$\neg (p \wedge q) \Leftrightarrow \neg p \vee \neg q$$

$$\neg (p \vee q) \Leftrightarrow \neg p \wedge \neg q$$

3. Luật giao hoán $p \vee q \Leftrightarrow q \vee p$

$$p \wedge q \Leftrightarrow q \wedge p$$

4. Luật kết hợp $(p \vee q) \vee r \Leftrightarrow p \vee (q \vee r)$

$$(p \wedge q) \wedge r <=> p \wedge (q \wedge r)$$

Các luật tương đương logic (cont)

5. Luật phân phối

$$p \lor (q \land r) \Leftrightarrow (p \lor q) \land (p \lor r)$$

$$p \land (q \lor r) \Leftrightarrow (p \land q) \lor (p \land r)$$

6. Luật lũy đẳng $p \lor p \Leftrightarrow p$

$$p \land p \Leftrightarrow p$$

7. Luật trung hòa $p \lor 0 \Leftrightarrow p$

$$p \land 1 \Leftrightarrow p$$

➢ Các luật tương đương logic (cont)

8. Luật về phần tử bù

$$p \wedge \neg p \Leftrightarrow 0$$

$$p \vee \neg p \Leftrightarrow 1$$

9. Luật thống trị

$$p \wedge 0 \Leftrightarrow 0$$

$$p \vee 1 \Leftrightarrow 1$$

10. Luật hấp thụ

$$\mathbf{p} \vee (\mathbf{p} \wedge q) \Leftrightarrow \mathbf{p}$$

$$\mathbf{p} \wedge (\mathbf{p} \vee q) \Leftrightarrow \mathbf{p}$$

Các luật tương đương logic (cont)

11. Luật về phép kéo theo:

$$p \rightarrow q \Leftrightarrow \neg p \vee q$$
$$\Leftrightarrow \neg q \rightarrow \neg p$$

Ví dụ: Nếu trời mưa thì đường trơn ⇔ nếu đường không trơn thì trời không mưa

➢ Các luật tương đương logic (cont)

VD: Cho p, q, r là các biến mệnh đề. Chứng minh rằng:

- $(\neg p \to r) \wedge (q \to r) \Leftrightarrow (p \to q) \to r$

Ta có: $(\neg p \to r) \wedge (q \to r)$

$\Leftrightarrow (p \vee r) \wedge (\neg q \vee r)$ (luật 11. về phép kéo theo)

$\Leftrightarrow (p \wedge \neg q) \vee r$ (luật phân phối)

$\Leftrightarrow \neg(\neg p \vee q) \vee r$ (De Morgan)

$\Leftrightarrow \neg(p \to q) \vee r$ (luật 11. về phép kéo theo)

$\Leftrightarrow (p \to q) \to r$ (luật 11. về phép kéo theo)

➢ Các luật tương đương logic (cont): VD

$$\overline{p \vee (\overline{p} \wedge q)} \Leftrightarrow \overline{p} \wedge \overline{q} \ ?$$

Chứng minh:

$\overline{p \vee (\overline{p} \wedge q)} \Leftrightarrow \overline{p} \wedge \overline{(\overline{p} \wedge q)}$ theo luật De Morgan thứ 2

$\Leftrightarrow \overline{p} \wedge \left[(\overline{\overline{p}}) \vee \overline{q} \right]$ theo luật De Morgan thứ 2

$\Leftrightarrow \overline{p} \wedge \left[p \vee \overline{q} \right]$ theo luật phủ định kép

$\Leftrightarrow (\overline{p} \wedge p) \vee (\overline{p} \wedge \overline{q})$ theo luật phân phối

$\Leftrightarrow F \vee (\overline{p} \wedge \overline{q})$

$\Leftrightarrow (\overline{p} \wedge \overline{q}) \vee F$

$\Leftrightarrow (\overline{p} \wedge \overline{q})$ Điều cần chứng minh.

➢ Dạng chuẩn tắc hội

- Một mệnh đề tuyển là tuyển của các mệnh đề nguyên thủy
  - Mệnh đề tuyển có dạng $p_1 \vee p_2 \vee \ldots \vee p_n$ trong đó p_i là các *mệnh đề nguyên thủy*
- Một công thức ở dạng chuẩn tắc hội nếu nó là hội của các mệnh đề tuyển

$$(a \vee e \vee f \vee g) \wedge (b \vee c \vee d)$$

➢ Dạng chuẩn tắc hội (cont)

- Có thể biến đổi một công thức bất kỳ về dạng chuẩn tắc hội bằng cách biến đổi theo nguyên tắc sau:
  - Khử các phép tương đương: $a \leftrightarrow b \equiv (a \to b) \land (b \to a)$
  - Khử các phép kéo theo: $a \to b \equiv \neg a \lor b$
  - Chuyển các phép phủ định vào sát các ký hiệu mệnh đề bằng cách áp dụng luật De Morgan
  - Khử phủ định kép: $\neg(\neg a) \equiv a$
  - Áp dụng luật phân phối: $a \lor (b \land c) \equiv (a \lor b) \land (a \lor c)$

➢ Dạng chuẩn tắc hội (cont): VD

VD: Chuẩn hóa về dạng chuẩn tắc hội:

$$(p \rightarrow q) \vee \neg(r \vee \neg s)$$

Giải

$$(p \rightarrow q) \vee \overline{(r \vee \overline{s})} \Leftrightarrow (\overline{p} \vee q) \vee \overline{(r \vee \overline{s})}$$

$$\Leftrightarrow (\overline{p} \vee q) \vee (\overline{r} \wedge \overline{\overline{s}})$$

$$\Leftrightarrow (\overline{p} \vee q) \vee (\overline{r} \wedge s)$$

$$\Leftrightarrow (\overline{p} \vee q \vee \overline{r}) \wedge (\overline{p} \vee q \vee s)$$

Bài tập 1

- Chứng minh các mệnh đề sau là *hằng đúng*.

a) $(p \wedge q) \to p$

b) $p \to (p \vee q)$

c) $\neg p \to (p \to q)$

d) $(p \wedge q) \to (p \to q)$

e) $\neg(p \to q) \to p$

f) $\neg(p \to q) \to \neg q$

g) $\neg p \wedge (p \vee q) \to q$

h) $(p \to q) \wedge (q \to r) \to (p \to r)$

i) $(p \wedge (p \to q)) \to q$

j) $((p \vee q) \wedge (p \to r) \wedge (q \to r)) \to r$

Bài tập 2

- Chứng minh các *tương đương logic* sau:

1) $(p \leftrightarrow q) \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$

2) $\neg p \leftrightarrow q \equiv p \leftrightarrow \neg q$

3) $\neg(p \leftrightarrow q) \equiv \neg p \leftrightarrow q$

Vị từ và lượng từ (Logic vị từ)

➢ Logic vị từ

1. Định nghĩa Vị từ là một khẳng định p(x,y,..), trong đó x,y...là các biến thuộc tập hợp A, B,… cho trước sao cho:

- Bản thân p(x,y,..) **không phải** là mệnh đề.
- Nếu thay x,y,… thành **giá trị cụ thể** thì p(x,y,..) là mệnh đề. Các biến x,y là các biến tự do của vị từ

Ví dụ. Các phát biểu sau là vị từ (chưa là mệnh đề)

- p(n) = "n +1 là số nguyên tố".
- q(x,y) = "$x^2 + y = 1$" .
- r(x,y,z) = "$x^2 + y^2 > z$".

Khi thay các giá trị cụ thể của n,x,y,z thì chúng là các MĐ.

- **Ví dụ: P(n) = {n là chẵn}**
  - n = 2: {2 là chẵn}: True
  - n = 5: {5 là chẵn}: False

➢ Vị từ (Hàm mệnh đề)

- Ví dụ:
  - Cho Q(x, y) là hàm mệnh đề xác định câu x= y+3. Hãy xác định giá trị chân lý của các mệnh đề Q(1,2) và Q(3,0)?
  - Q(1,2) là **sai**. Q(3,0) là **đúng**.

➢ Vị từ (Hàm mệnh đề) (cont)

- **Không gian của vị từ:** có thể xem vị từ như là một ánh xạ P, $\forall$ $x \in E$ ta được một ảnh $P(x) \in \{0, 1\}$. Tập hợp E này được gọi là không gian của vị từ.
- **Trọng lượng của vị từ:** số biến của vị từ
- **Ví dụ:**
  - P(a,b) = {cặp số nguyên tương ứng thỏa a + b = 5}
  - Không gian của vị từ: Số nguyên
  - Trọng lượng: 2

➢ Các phép toán trên vị từ

Cho trước các vị từ p(x), q(x) theo một biến $x \in A$. Khi ấy, ta cũng có các phép toán tương ứng như trên mệnh đề

- *Phủ định* $\neg p(x)$
- *Phép nối liền (hội)* $p(x) \wedge q(x)$
- *Phép nối rời (tuyển)* $p(x) \vee q(x)$

-*Phép XOR* $p(x) + q(x)$

- *Phép kéo theo* $p(x) \rightarrow q(x)$
- *Phép kéo theo hai chiều (tương đương)* $p(x) \leftrightarrow q(x)$

➢ Các phép toán trên vị từ (cont)

- **Định nghĩa:** Cho P(x) là một vị từ có không gian là A. Các mệnh đề **lượng tử** hóa (quantified statement) của P(x) như sau:
  - Mệnh đề "***Với mọi x thuộc A, P(x)*** ", kí hiệu bởi

    $$\text{“}\forall x \in A, P(x)\text{”},$$

    là mệnh đề đúng ⇔ P(a) luôn đúng với mọi giá trị a ∈ A.
  - Mệnh đề "***Tồn tại một x thuộc A, P(x))***" kí hiệu bởi :

    $$\text{“}\exists x \in A, P(x)\text{”},$$

    là mệnh đề đúng khi và chỉ khi có một giá trị x = a nào đó sao cho mệnh đề P(a) đúng.

➢ Các phép toán trên vị từ (cont): VD

- Không lập bảng chân trị, sử dụng các tương đương logic để chứng minh rằng :

$(P \wedge Q) \rightarrow Q$ là hằng đúng.

$$
\begin{aligned}
(P \wedge Q) \rightarrow Q &= \overline{P \wedge Q} \vee Q \\
&= (\overline{P} \vee \overline{Q}) \vee Q \\
&= \overline{P} \vee (\overline{Q} \vee Q) \\
&= \overline{P} \vee T \\
&= T
\end{aligned}
$$

➢ Các phép toán trên vị từ (cont): VD

- Bằng biến đổi tương đương, chứng minh các biểu thức mệnh đề sau là hằng đúng
  - **$(P \wedge Q) \rightarrow P$**

$$
\begin{aligned}
(P \wedge Q) \rightarrow P &= \neg (P \wedge Q) \vee P \\
&= (\neg P \vee \neg Q) \vee P \\
&= \neg P \vee P \vee \neg Q \\
&= T \vee \neg Q = T
\end{aligned}
$$

  - **$P \rightarrow (\neg P \rightarrow P)$**

$$
\begin{aligned}
P \rightarrow (P \vee P) &= P \rightarrow P \\
&= \neg P \vee P = T
\end{aligned}
$$

➢ Các phép toán trên vị từ (cont): VD

- Bằng biến đổi tương đương, chứng minh các biểu thức mệnh đề sau là hằng đúng
- **$P \rightarrow (Q \rightarrow (P \wedge Q))$**

$$
\begin{aligned}
P \rightarrow (\neg Q \text{ v } (P \wedge Q)) &= P \rightarrow ((\neg Q \text{v} P) \wedge (\neg Q \text{v} Q) \\
&= P \rightarrow ((\neg Q \text{v} P) \wedge T \\
&= P \rightarrow ((\neg Q \text{v} P) \\
&= \neg P \text{v } (\neg Q \text{v} P) \\
&= \neg P \text{ v } \neg Q \text{ v } P \\
&= \neg P \text{ v } P \text{ v } \neg Q \\
&= T \text{ v } \neg Q = T \text{ (hằng đúng)}
\end{aligned}
$$

➤ Các phép toán trên vị từ (cont): VD

- Chứng minh biểu thức mệnh đề $(\neg((r \vee q) \wedge q) \vee \neg p) \wedge ((\neg p \vee \neg q) \rightarrow (p \wedge q \wedge r))$ là **hằng sai**

$$\neg((r \vee q) \wedge q) \vee \neg p \equiv \neg q \vee \neg p \equiv \neg(p \wedge q)$$

$$(\neg p \vee \neg q) \rightarrow (p \wedge q \wedge r) \equiv \neg(\neg p \vee \neg q) \vee (p \wedge q \wedge r)$$

$$\equiv (p \wedge q) \vee (p \wedge q \wedge r)$$

$$\equiv p \wedge q$$

$$(\neg((r \vee q) \wedge q) \vee \neg p) \wedge ((\neg p \vee \neg q) \rightarrow (p \wedge q \wedge r))$$

$$\equiv \neg(p \wedge q) \wedge (p \wedge q) \equiv F$$

Các phép toán trên vị từ (cont): VD

- Chứng minh biểu thức mệnh đề sau là hằng sai

$\neg p \wedge \neg(p \wedge q) \wedge \neg(p \wedge \neg r) \wedge (((\neg q \rightarrow r) \vee \neg(q \vee (r \wedge s) \vee (r \wedge \neg s))) \wedge p)$

$\neg p \wedge \neg(p \wedge q) \wedge \neg(p \wedge \neg r) \equiv \neg(p \vee (p \wedge q) \vee (p \wedge \neg r))$

$\equiv \neg \mathbf{p}$

$((\neg q \rightarrow r) \vee \neg(q \vee (r \wedge s) \vee (r \wedge \neg s))) \wedge p$

$\equiv ((q \vee r) \vee \neg(q \vee (r \wedge (s \vee \neg s)))) \wedge p$

$\equiv ((q \vee r) \vee \neg(q \vee r)) \wedge p \equiv \mathbf{p}$

$\neg(p \vee (p \wedge q) \vee (p \wedge \neg r)) \vee (p \wedge ((\neg q \rightarrow r) \vee \neg(q \vee (r \wedge s) \vee (r \wedge \neg s)))) \equiv \neg p \wedge p \equiv F$

➤ Các phép toán trên vị từ (cont): VD

Chứng minh biểu thức mệnh đề $\neg(p \vee q) \vee (\neg p \wedge q) \wedge \neg q$ tương đương với biểu thức $(p \rightarrow q) \wedge (\neg q \wedge (r \vee \neg q))$

$$\neg(p \vee q) \vee (\neg p \wedge q) \wedge \neg q \equiv (\neg p \wedge \neg q) \vee (\neg p \wedge q) \wedge \neg q$$

$$\equiv (\neg p \wedge (\neg q \vee q)) \wedge \neg q$$

$$\equiv \neg p \wedge \neg q \ \mathbf{(1)}$$

$$(p \rightarrow q) \wedge (\neg q \wedge (r \vee \neg q)) \equiv (\neg p \vee q) \wedge \neg q$$

$$\equiv (\neg p \wedge \neg q) \vee (q \wedge \neg q)$$

$$\equiv (\neg p \wedge \neg q) \vee F \equiv \neg p \wedge \neg q \quad \mathbf{(2)}$$

(1) & (2) $\Rightarrow \neg(p \vee q) \vee (\neg p \wedge q) \wedge \neg q$ tương đương

$$(p \rightarrow q) \wedge (\neg q \wedge (r \vee \neg q))$$

➢ Các phép toán trên vị từ (cont): VD

Chứng minh biểu thức mệnh đề $\neg(\neg((r \vee q) \wedge q) \vee \neg p)$ tương đương với biểu thức $(\neg p \vee \neg q) \rightarrow (p \wedge q \wedge r)$

$$\boldsymbol{\neg(\neg((r \vee q) \wedge q) \vee \neg p)} \equiv \neg(\neg q \vee \neg p)$$

$$\equiv p \wedge q \qquad \textbf{(1)}$$

$$\boldsymbol{(\neg p \vee \neg q) \rightarrow (p \wedge q \wedge r)} \equiv \neg(\neg p \vee \neg q) \vee (p \wedge q \wedge r)$$

$$\equiv (p \wedge q) \vee (p \wedge q \wedge r)$$

$$\equiv p \wedge q \qquad \textbf{(2)}$$

$\textbf{(1)}\&\ (2) \Rightarrow \neg(\neg((r \vee q) \wedge q) \vee \neg p)$ tương đương

$$(\neg p \vee \neg q) \rightarrow (p \wedge q \wedge r)$$

Các phép toán trên vị từ (cont): VD

Chứng minh biểu thức mệnh đề $p \vee ((p \wedge q) \vee (p \wedge \neg r))$ tương đương với biểu thức mệnh đề $p \wedge ((\neg q \rightarrow r) \vee \neg(q \vee (r \wedge s) \vee (r \wedge \neg s)))$

$\mathbf{p \vee ((p \wedge q) \vee (p \wedge \neg r))} \equiv p \vee (p \wedge (q \vee \neg r)) \equiv p \quad \mathbf{(1)}$

$\mathbf{p \wedge ((\neg q \rightarrow r) \vee \neg(q \vee (r \wedge s) \vee (r \wedge \neg s)))}$

$\equiv p \wedge ((q \vee r) \vee \neg(q \vee (r \wedge (s \vee \neg s))))$

$\equiv \quad p \wedge ((q \vee r) \vee \neg(q \vee r)) \equiv p \quad \mathbf{(2)}$

(1) & (2) $\Rightarrow$ hai mệnh đề là tương đương

➢ Logic vị từ: Tóm tắt

Khi xét một mệnh đề p(x) với x $\in$ A. Ta có các trường hợp sau

- TH1. Khi thay x bởi 1 phần tử a tùy ý$\in$ A, ta có p(a) đúng.
- TH2. Với một số giá trị a $\in$ A, ta có p(a) đúng.
- TH3. Khi thay x bởi 1 phần tử a tùy ý$\in$ A, ta có p(a) sai.

Ví dụ. Cho các vị từ p(x) sau với x$\in$**R**

- p(x) = "x^2 +1 >0" đúng với x tuỳ ý (với mọi x).
- p(x) = "x^2 -2x+1=0" chỉ đúng với x = 1.
- p(x) = "x^2 -2x+3=0" sai với x tuỳ ý (với mọi x).

➢ Lượng từ

- Là phương pháp để biến một vị từ (hàm mệnh đề) thành một mệnh đề mà *không cần phải kiểm chứng, mọi giá trị chân lý* của hàm mệnh đề tương ứng với các giá trị của biến thuộc miền đang xét.
- Hai lượng từ quan trọng:
  - lượng từ với mọi ($\forall$),
  - lượng từ tồn tại ($\exists$).

➢ Lượng từ (cont)

Định nghĩa. Cho p(x) là một vị từ theo một biến xác định trên A. Ta định nghĩa các mệnh đề lượng từ hóa của p(x) như sau:

- Mệnh đề *"Với mọi x thuộc A, p(x)* ", kí hiệu bởi

"$\forall$x $\in$ A, p(x)",

là mệnh đề đúng khi và chỉ khi p(a) luôn đúng với mọi giá trị a $\in$ A.

- Mệnh đề "*Tồn tại (ít nhất) hay có (ít nhất) một x thuộc A, p(x)"* kí hiệu bởi **:**

"$\exists$x $\in$ A, p(x)" ,

là mệnh đề đúng khi và chỉ khi có ít nhất một giá trị x = a_0 nào đó sao cho mệnh đề p(a_0) đúng.

➢ Lượng từ (cont)

$\forall$: *được gọi là lượng từ* ***phổ dụng (với mọi)***

$\exists$: *được gọi là lượng từ* ***tồn tại***

Ví dụ. Các mệnh đề sau đúng hay sai

- "$\forall x \in \mathbf{R}, x^2 + 3x + 1 \leq 0$"
- "$\exists x \in \mathbf{R}, x^2 + 3x + 1 \leq 0$"
- "$\forall x \in \mathbf{R}, x^2 + 1 \geq 2x$"
- "$\exists x \in \mathbf{R}, x^2 + 1 < 0$"

➢ Mệnh đề lượng từ hoá

Định nghĩa. Cho p(x, y) là một vị từ theo hai biến x, y xác định trên $A \times B$. Ta định nghĩa các mệnh đề lượng từ hóa của p(x, y) như sau:

"$\forall x \in A, \forall y \in B, p(x, y)$" = "$\forall x \in A, (\forall y \in B, p(x, y))$"

"$\forall x \in A, \exists y \in B, p(x, y)$" = "$\forall x \in A, (\exists y \in B, p(x, y))$"

"$\exists x \in A, \forall y \in B, p(x, y)$" = "$\exists x \in A, (\forall y \in B, p(x, y))$"

"$\exists x \in A, \exists y \in B, p(x, y)$" = "$\exists x \in A, (\exists y \in B, p(x, y))$"

➢ Mệnh đề lượng từ hoá: (cont): VD 1

- Mệnh đề "**$\forall x \in R, \forall y \in R, x + 2y < 1$**" đúng hay sai?

Mệnh đề sai vì tồn tại $x_0 = 0$, $y_0 = 1 \in R$ mà $x_0 + 2y_0 \geq 1$.

- Mệnh đề "**$\forall x \in R, \exists y \in R, x + 2y < 1$**" đúng hay sai?

Mệnh đề đúng vì với mỗi $x = a \in R$, tồn tại $y_a \in R$ như

$y_a = -a/2$, sao cho $a + 2y_a < 1$.

➢ Mệnh đề lượng từ hoá: (cont) VD 2

- Mệnh đề "**$\exists x \in R, \forall y \in R, x + 2y < 1$**" đúng hay sai

Mệnh đề sai vì không thể có $x = a \in R$ để bất đẳng thức $a + 2y < 1$ được thỏa với mọi $y \in R$ (chẳng hạn, $y = -a/2 + 2$ không thể thỏa bất đẳng thức này).

- Mệnh đề "**$\exists x \in R, \exists y \in R, x + 2y < 1$" đúng hay sai?**

Mệnh đề đng vì tồn tại $x_0 = 0$, $y_0 = 0 \in$ **R** chẳng hạn thỏa $x_0 + 2y_0 < 1$.

➢ Logic lượng từ

Định lý. Cho p(x, y) là một vị từ theo hai biến x, y xác định trên $A \times B$. Khi đó:

1) "$\forall x \in A, \forall y \in B, p(x, y)$" $\Leftrightarrow$ "$\forall y \in B, \forall x \in A, p(x, y)$"

2) "$\exists x \in A, \exists y \in B, p(x, y)$" $\Leftrightarrow$ "$\exists y \in B, \exists x \in A, p(x, y)$"

3) "$\exists x \in A, \forall y \in B, p(x, y)$" $\Rightarrow$ "$\forall y \in B, \exists x \in A, p(x, y)$"

Chiều đảo của 3) nói chung không đúng.

➢ Phủ định của mệnh đề lượng từ

Phủ định của mệnh đề lượng từ hóa vị từ p(x,y,..) có được bằng các thay ∀ thành ∃, thay ∃ thành ∀ và vị từ p(x,y,..) thành ¬ p(x,y,..).

Với vị từ theo 1 biến ta có :

$$\overline{\forall x \in A, p(x)} \Leftrightarrow \exists x \in A, \overline{p(x)}$$

$$\overline{\exists x \in A, p(x)} \Leftrightarrow \forall x \in A, \overline{p(x)}$$

➢ Phủ định của mệnh đề lượng từ: (cont)

Với vị từ theo 2 biến.

$$\overline{\forall x \in A, \forall y \in B, p(x, y)} \Leftrightarrow \exists x \in A, \exists y \in B, \overline{p(x, y)}$$

$$\overline{\forall x \in A, \exists y \in B, p(x, y)} \Leftrightarrow \exists x \in A, \forall y \in B, \overline{p(x, y)}$$

$$\overline{\exists x \in A, \forall y \in B, p(x, y)} \Leftrightarrow \forall x \in A, \exists y \in B, \overline{p(x, y)}$$

$$\overline{\exists x \in A, \exists y \in B, p(x, y)} \Leftrightarrow \forall x \in A, \forall y \in B, \overline{p(x, y)}$$

➢ Phủ định của mệnh đề lượng từ (cont)

Ví dụ phủ định các mệnh đề sau

- "$\forall x \in A,\ 2x + 1 \leq 0$"

- "$\forall \varepsilon > 0, \exists \delta > 0, \forall x \in R, |x - a| < \delta \rightarrow |f(x) - f(a)| < \varepsilon$".

Trả lời

"$\exists x \in A,\ 2x + 1 > 0$"

"$\exists \varepsilon > 0, \forall \delta > 0, \exists x \in R, |x - a| < \delta \wedge (|f(x) - f(a)| \geq \varepsilon)$".

➢ Đặc biệt hóa phổ dụng

Qui tắc đặc biệt hóa phổ dụng:

Nếu một mệnh đề đúng có dạng lượng từ hóa trong đó một biến x $\in$ A bị buộc bởi lượng từ phổ dụng $\forall$, khi ấy nếu thay thế x bởi a $\in$ A ta sẽ được một mệnh đề đúng

$$\begin{array}{l} \forall x \in A,\ p(x) \\ a \in A \\ \hline \therefore\ p(a) \end{array}$$

Ví dụ:

“Mọi người đều chết”
“Socrate là người”
Vậy “Socrate cũng chết”

➢ Nhắc lại lượng từ

Giá trị chân lý của lượng từ ∀, ∃

∀x P(x)	P(x) đúng với mọi x	Có một giá trị của x để P(x) sai
∃x P(x)	Có một giá trị của x để P(x) đúng	P(x) sai với mọi x

- **Ứng dụng:**
  - Dịch những câu thông thường thành biểu thức logic.

➢ Dịch những câu thông thường thành biểu thức logic

- <u>Ví dụ 1</u>:
  - "*Bạn không được lái xe máy nếu bạn cao dưới 1.5 mét trừ phi bạn trên 18 tuổi*".
- <u>Lời giải</u>:
  - p : "Bạn được lái xe máy".
  - q : "Bạn cao dưới 1.5m".
  - r : "Bạn trên 18 tuổi".
  - => Câu hỏi trên được dịch là: $(q \wedge \overline{r}) \rightarrow \overline{p}$

➢ Dịch những câu thông thường thành biểu thức logic (cont)

- Ví dụ 2:
  - “*Tất cả các sinh viên học tin học đều học môn toán học rời rạc*”.
- Lời giải: Gọi
  - P(x) là câu “*x* cần học môn toán học rời rạc”,
  - *x* được xác định trong miền không gian của các sinh viên học tin học.
  - Khi đó chúng ta có thể phát biểu: $\forall x\ P(x)$.

➢ Dịch những câu thông thường thành biểu thức logic (cont)

- Ví dụ 3:
  - “*Có một sinh viên ở lớp này ít nhất đã ở tất cả các phòng của ít nhất một nhà trong ký túc xá*”.
- Lời giải:
  - tập sinh viên trong lớp là không gian xác định sinh viên *x*,
  - tập các nhà trong ký túc xá là không gian xác định căn nhà *y*,
  - tập các phòng là không gian xác định phòng *z*.
  - P(z,y) là “*z thuộc y*”, Q(x,z) là “*x đã ở z*”.
  - Khi đó chúng ta có thể phát biểu:

$$\exists x \exists y \forall z \ (P(z,y) \rightarrow Q(x,z))$$

➢ Bài tập

Dịch câu thành biểu thức logic

- BT1: "Mọi người đều có chính xác một người bạn tốt nhất“.
- BT2: "Nếu một người nào đó là phụ nữ và đã sinh con, thì người đó sẽ là mẹ của một người nào khác“.
- BT3: Xét các câu sau. Hai câu đầu tiên là tiền đề và câu ba là kết luận. Toàn bộ tập hợp 3 câu này được gọi là một suy lý.
  - "Tất cả sư tử Hà Đông đều hung dữ".
  - "Một số sư tử Hà Đông không uống cà phê".
  - "Một số sinh vật hung dữ không uống cà phê".