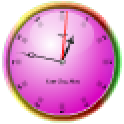


Toán rời rạc 1

Discrete mathematics 1

Bài 2: Một số kiến thức cơ sở

Nội dung Bài 2

1. Lý thuyết tập hợp
2. Logic mệnh đề
3. Logic vị từ
4. Thuật toán và độ phức tạp
5. Bài tập

Lý thuyết tập hợp

- Một cách trực quan (Kenneth H. Rosen 2018):
  - A set is an unordered collection of distinct objects, called elements or members of the set. A set is said to contain its elements. We write a ∈ A to denote that a is an element of the set A. The notation a ∉ A denotes that a is not an element of the set A.
- Số phần tử của tập hợp A: $|A|$
- Một tập hợp có n phần tử được gọi là một n-tập; n được gọi là bản số của tập
- Tập hợp con: $A \subseteq B$ ($x \in A \Rightarrow x \in B$)
- Tập hợp bằng nhau : nếu $A \subseteq B$ và $B \subseteq A$ thì $A = B$
- Tập rỗng : ∅ (Không có phần tử nào, là con của mọi tập hợp)

Các phép toán trên tập hợp

- Phần bù của A trong X: $\bar{A} = \{x \in X \mid x \notin A\}$
- Hợp của hai tập hợp: $A \cup B = \{x \mid x \in A \text{ hoặc } x \in B\}$
- Giao của hai tập hợp: $A \cap B = \{x \mid x \in A \text{ và } x \in B\}$
- Hiệu của hai tập hợp: $A \setminus \mathrm{B} = \{x \mid x \in A \text{ và } x \notin B\}$
- Luật kết hợp: $(A \cup B) \cup C = A \cup (B \cup C)$; $(A \cap B) \cap C = A \cap (B \cap C)$
- Luật giao hoán: $A \cup B = B \cup A$, $A \cap B = B \cap A$
- Luật phân bố: $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

  $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
- Luật đối ngẫu: $\overline{A \cup B} = \bar{A} \cap \bar{B}$; $\overline{A \cap B}$ ष $\bar{A} \cup \bar{B}$
- Tích Đề các: $A \times B = \{(a,b) \mid a \in A,\ b \in B\}$

Quan hệ

- Quan hệ: Một quan hệ hai ngôi R trên tập hữu hạn phần tử X được định nghĩa như là
tập con R(X) của tích Đề các X×X.
- Một số tính chất có thể có của một quan hệ R:
  - Phản xạ: mọi phần tử có quan hệ với chính nó
  - Đối xứng: *a* có quan hệ với *b* kéo theo *b* có quan hệ với *a*
  - Kéo theo: *a* có quan hệ với *b* và *b* có quan hệ với *c* kéo theo *a* có quan hệ với *c*
- Ví dụ:

  $X = \{1, 2, 3, 4\}$

  $a, b \in X$, *a* có quan hệ *R* đối với *b* nếu *a* chia hết cho *b*

  $R(X) = \{(1,1), (2,1), (2,2), (3,1), (3,3), (4,1), (4,2), (4,4)\} \subset X \times X$

  - R(X): Phản xạ, kéo theo, nhưng không đối xứng

Quan hệ tương đương và phân hoạch

- Quan hệ tương đương: là một quan hệ có đủ ba tính chất, phản xạ, đối xứng, và kéo theo
- Lớp tương đương: một quan hệ tương đương trên tập hợp sẽ chia tập hợp thành các lớp tương đương sao cho
  - Hai phần tử thuộc cùng một lớp có quan hệ với nhau
  - Hai phần tử khác lớp không có quan hệ với nhau
  - Các lớp tương đương phủ kín tập hợp ban đầu
- Phân hoạch: là một họ các lớp tương đương (các tập con khác rỗng, có các tính chất trên) của một tập hợp
- Một quan hệ tương đương trên tập hợp sẽ xác định một phân hoạch trên tập hợp. Ngược lại một phân hoạch bất kỳ trên tập hợp sẽ tương ứng với một quan hệ tương đương trên nó.

Ví dụ 1 về quan hệ tương đương

- Cho X = {Tập các sinh viên của một lớp}
- Gọi R(X) = {(a,b) | a và b là các sinh viên cùng họ};
- Có thể thấy: R là quan hệ tương đương.
- Thật vậy:
  - R phản xạ vì: a R a hay a luôn luôn cùng họ với a
  - R đối xứng vì: a R b thì b R a hay nếu a cùng họ với b thì b cũng cùng họ với a
  - R bắc cầu vì: a R b và b R c thì a R c vì a và c chắc chắn có cùng họ
- Chú ý: Nếu cho quan hệ R trên X hay R(X) cho dưới dạng tập hợp thì việc xét các tính chất dựa trên việc kiểm tra quan hệ a R b chính là kiểm tra (a,b) có thuộc R(X) không.

Ví dụ 2 về quan hệ tương đương

- Xét $X = \{1, 2, \ldots, m\}$, m là số nguyên dương ($m > 2$), k là số nguyên dương, $1 < k < m$
- Định nghĩa quan hệ R trên X như sau:
  - Với $a, b \in X$: aRb $\Leftrightarrow a \equiv b (mod\ k)$
    - a có quan hệ R với b nếu a và b có cùng số dư khi chia cho k hay $a \equiv b (mod\ k)$
  - R là quan hệ tương đương (Phản xạ, đối xứng, kéo theo)
- Đặt $A_i = \{a \in X \mid a \equiv i\ (mod\ k)\}$, $i = 0, 1, \ldots, k-1$
  - $A_0, A_1, \ldots, A_{k-1}$ tạo thành một phân hoạch của X
  - Một cách tương đương: $A_i \cap A_j = \emptyset, i \neq j$ và $X = \bigcup_{i=0}^{k-1} A_i$

Quan hệ R ở ví dụ này gọi là quan hệ đồng dư

Ví dụ 2 về quan hệ tương đương

- Ví dụ cụ thể: X = {1, 2, 3, 4, 5, 6}, $m = 6$ ($m > 2$); xét $k = 3$ ($1 < k < 6$)
  - Xét quan hệ R: $a, b \in X$: aRb $\Leftrightarrow a \equiv b(mod\ 3)$ hay a và b có cùng số dư khi chia cho 3 hay $a \equiv b(mod\ 3)$
  - Khi đó các tập: $A_i = \{a \in X \mid a \equiv i\ (mod\ 3)\}$, $i = 0, 1, \ldots, 2$ như sau:
    - $A_0 = \{a \in X \mid a \equiv 0\ (mod\ 3)\} = \{3, 6\}$, $i = 0$
    - $A_1 = \{a \in X \mid a \equiv 1\ (mod\ 3)\} = \{1, 4\}$, $i = 1$
    - $A_2 = \{a \in X \mid a \equiv 2\ (mod\ 3)\} = \{2, 5\}$, $i = 2$
  - Từ A_0, A_1 và A_2 có thể thấy rằng: $A_i \cap A_j = \emptyset, i \neq j$ và $X = \bigcup_{i=0}^{k-1} A_i$

Nguyên lý cộng

- Nếu A và B là hai tập rời nhau thì: $|A \cup B| = |A| + |B|$
- Nếu $\{A_1, A_2, \ldots, A_k\}$ là một phân hoạch của tập hợp X thì: $|X| = |A_1| + |A_2| + \cdots + |A_k|$
- Nếu A là một tập con của X thì: $|\bar{A}| = |X| - |A|$
- Ví dụ:
  - Một đoàn vận động viên - VĐV thi 2 môn bắn súng và bơi. Số VĐV nam: 10. Số VĐV thi bắn súng: 14. Số nữ VĐV thi bơi = số nam VĐV thi bắn súng. Toàn đoàn có bao nhiêu VĐV?

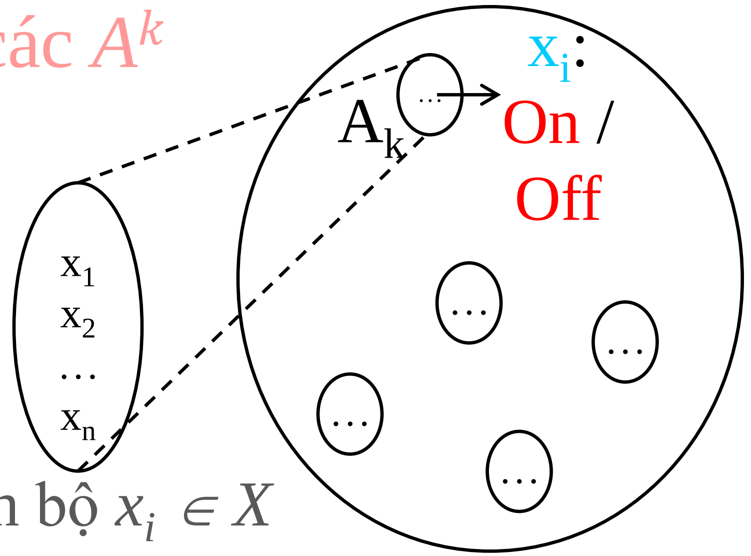
Bắn súng nữ + Bơi nữ = Bắn súng (nữ + nam) = 14 → Tổng: 14 + 10 = 24

Nguyên lý nhân

- Nếu mỗi thành phần a_i của bộ có thứ tự k thành phần $(a_1, a_2, \ldots, a_k)$ có n_i khả năng chọn, thì số bộ được tạo ra sẽ là tích các khả năng $n_1 n_2 \ldots n_k$
- Hệ quả:
  - $|A_1 \times A_2 \times \cdots \times A_k| = |A_1||A_2|\ldots|A_k|$
  - $|A^k| = |A|^k$
- Ví dụ
  - Có bao nhiêu số nguyên dương có 5 chữ số được thành lập bởi các chữ số 0, 1, 2?
  - Áp dụng nguyên lý nhân:
    - Chữ số hàng chục nghìn có 2 cách chọn, các chữ số còn lại có 3 cách chọn. Số các số: 2 × 3 × 3 × 3 × 3 = 162 số.

Chỉnh hợp lặp

- Định nghĩa: Một chỉnh hợp lặp chập k của n phần tử là một bộ có thứ tự gồm k thành phần, lấy từ n thành phần đã cho. Các thành phần có thể được lặp lại.
- Mỗi thành phần trong k thành phần có n lựa chọn, theo nguyên lý nhân, số chỉnh hợp lặp chập k của n phần tử là $n^k \leftrightarrow$ số phần tử của tích Đề các A^k
- Ví dụ
  - Tính số tập con của một n-tập?
  - X có n phần tử
  - Các tập con A_k chứa ít nhất 1 phần tử $x_i \in X$, nhiều nhất là toàn bộ $x_i \in X$
  - Mỗi x_i có 2 lựa chọn → có 2^n tập con A_k.



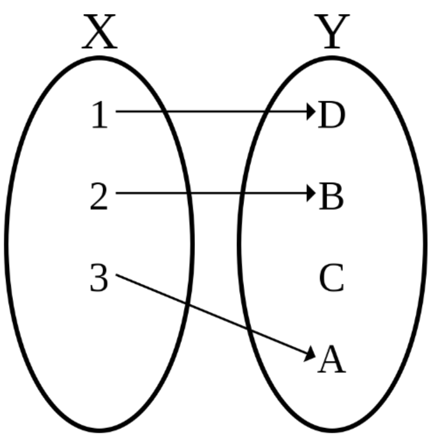
Tập lũy thừa của X

Chỉnh hợp không lặp – permutation

- Định nghĩa: Một chỉnh hợp không lặp chập k của n phần tử là một bộ có thứ tự gồm k thành phần, lấy từ $n$thành phần đã cho. Các thành phần không được lặp lại.
- Theo nguyên lý nhân, số chỉnh hợp không lặp chập k của n phần tử:

$$P(n,k) = n(n-1)..(n-k+1) = \frac{n!}{(n-k)!}$$

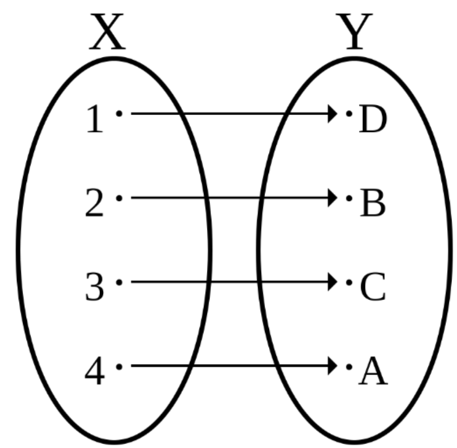
- Ví dụ
  - Tính số đơn ánh từ một k-tập vào một n-tập?
  - Nhắc lại: Đơn ánh
    - Xây dựng từ thành phần đầu tiên - n khả năng chọn. Mỗi thành phần tiếp theo những khả năng chọn giảm đi 1 (vì không được lấy lặp lại).
    - Tới thành phần thứ k có n - k + 1 khả năng chọn.



Đơn ánh

Hoán vị

- Định nghĩa: Ta gọi một hoán vị của n phần tử là một cách xếp thứ tự các phần tử đó
- Hoán vị ↔ song ánh của một tập n phần tử lên chính nó.
- Một hoán vị của n phần tử là một trường hợp riêng của chỉnh hợp không lặp khi $k = n$
- Số hoán vị của n phần tử: $n \times n - 1 \ldots 2 \times 1 = n!$
- Ví dụ
  - Cần bố trí việc thực hiện n chương trình trên một máy vi tính.
    Biết rằng các chương trình thực hiện độc lập (không phụ thuộc thứ tự).
    Hỏi có bao nhiêu cách?



Song ánh

Tổ hợp - combination

- Định nghĩa: Một tổ hợp chập k của n phần tử là một bộ không kể thứ tự gồm k thành phần khác nhau lấy từ $n$phần tử đã cho.
- Ký hiệu số tổ hợp: C_n^k ta có:

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

- Một số tính chất:

$$C_n^k = C_n^{n-k}$$

$$C_n^0 = C_n^n = 1$$

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

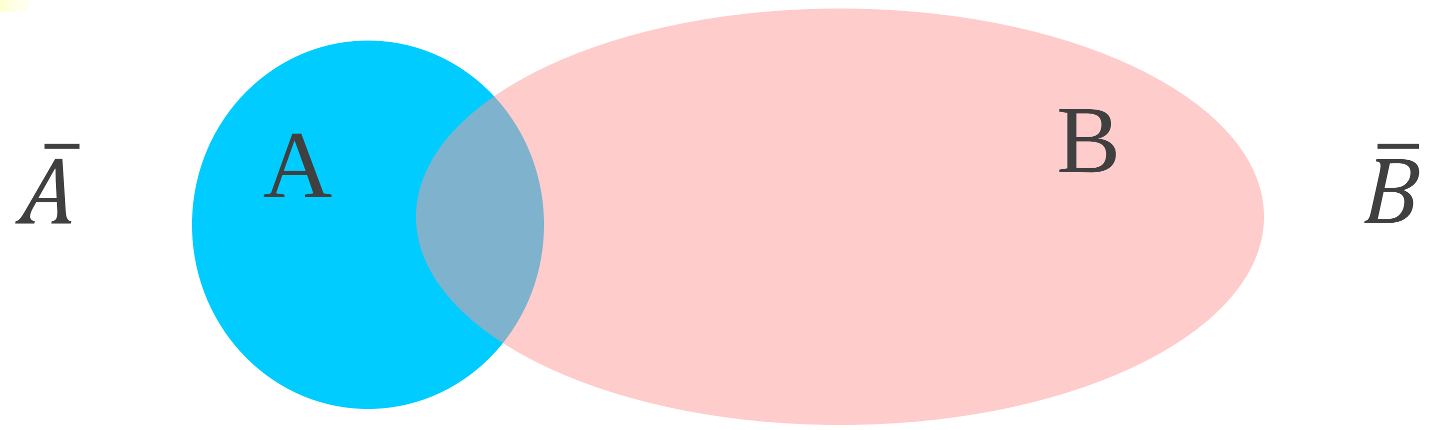
- Nhị thức Newton:

$$(x+y)^n = C_n^0 x^n + C_n^1 x^{n-1} y + \cdots + C_n^n y^n = \sum_{k=0}^{n} C_n^k x^{n-k} y^k$$

Bài tập 1 (Tập hợp)

1. Sử dụng định nghĩa chứng minh một số phép toán trên tập hợp
   - Ví dụ: Sử dụng định nghĩa chứng minh luật đối ngẫu hay luật De Morgan: $\overline{A \cup B} = \bar{A} \cap \bar{B}$; $\overline{A \cap B}$ ष $\bar{A} \cup \bar{B}$
   - …

Bài tập 1 (Tập hợp)



- Chứng minh: Gồm 2 bước (ví dụ chứng minh $\overline{A \cup B} = \bar{A} \cap \bar{B}$)
  - Bước 1: Chứng minh $\overline{A \cup B} \subseteq \bar{A} \cap \bar{B}$ (1)
  - Bước 2: Chứng minh $\bar{A} \cap \bar{B} \subseteq \overline{A \cup B}$ (2)
  - Chứng minh (1):
    Lấy x $\in \overline{A \cup B}$ $\rightarrow$ x $\notin A \cup B$ $\rightarrow$ x $\notin A$ và x $\notin B$ $\rightarrow$ x $\in \bar{A}$ và x $\in \bar{B}$
    $\rightarrow$ x $\in \bar{A} \cap \bar{B}$ hay $\overline{A \cup B} \subseteq \bar{A} \cap \bar{B}$
  - Chứng minh (2):
    Lấy x $\in \bar{A} \cap \bar{B}$ $\rightarrow$ x $\in \bar{A}$ và x $\in \bar{B}$ $\rightarrow$ x $\notin A$ và x $\notin B$ x $\notin A \cup B$
    $\rightarrow$ x $\in \overline{A \cup B}$ hay $\bar{A} \cap \bar{B} \subseteq \overline{A \cup B}$
  - Từ (1) và (2) suy ra đpcm: $\overline{A \cup B} = \bar{A} \cap \bar{B}$

Bài tập 2 (Tập hợp)

- Cho biết trong các hệ thức dưới đây hệ thức nào là đúng, hệ thức nào là sai?

  1) $A \subseteq A \cap B$

  2) $C \subseteq A \cap B \cup C$

  3) $A \cup B \subseteq A \cap B$

  4) $A \cap B \cup A = A \cap B$

  5) $(A \cup B)\backslash(A \cap B) = A\backslash \mathrm{B}$

Bài tập 2 (Tập hợp)

- Cho biết trong các hệ thức dưới đây hệ thức nào là đúng, hệ thức nào là sai?

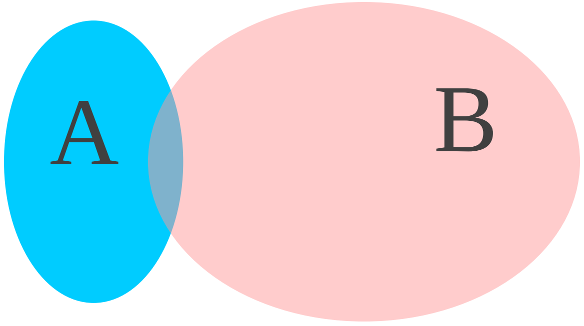
  1) $A \subseteq A \cap B$

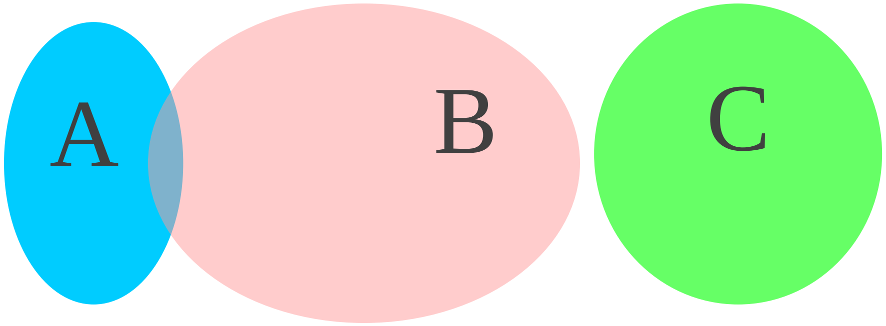
  2) $C \subseteq A \cap B \cup C$

  3) $A \cup B \subseteq A \cap B$

  4) $A \cap B \cup A = A \cap B$

  5) $(A \cup B) \backslash (A \cap B) = A \backslash B$





Bài tập 3 (Tập hợp)

- Ký hiệu Z là tập hợp các số nguyên.
  Xét hai tập con của Z:
  - $A = \{x \in Z \mid x = 4p - 1 \text{ với một } p \text{ nào đó} \in Z\}$
  - $B = \{y \in Z \mid y = 4q + 3 \text{ với một } q \text{ nào đó} \in Z\}$

  Chứng minh rằng $A = B$?

Bài tập 3 (Tập hợp)

- Ký hiệu Z là tập hợp các số nguyên.
  Xét hai tập con của Z:
  - $A = \{x \in Z \mid x = 4p - 1 \text{ với một } p \text{ nào đó} \in Z\}$
  - $B = \{y \in Z \mid y = 4q + 3 \text{ với một } q \text{ nào đó} \in Z\}$

  Chứng minh rằng $A = B$?
  - Chứng minh:
    - Lấy x bất kỳ ∈ A ta luôn chứng minh được rằng x ∈ B nên $A \subseteq B$.
      Thật vậy: x ∈ A → x = 4p – 1 với p là giá trị nào đó ∈ Z.
      Viết lại x như sau: x = 4p – 1 = 4p – 4 + 3 = 4(p-1) + 3.
      Do p ∈ Z nên p-1 ∈ Z từ đó suy ra z = 4(p-1) + 3 ∈ B
      Như vậy ta chứng minh được điều thứ nhất: $A \subseteq B$
    - Hoàn toàn tương tự chúng ta chứng minh được chiều ngược lại: B ⊆ A từ đó suy ra: A = B.

Bài tập 4 (Tập hợp)

- Cho $A = \{0, 1, 2, 3, 4\}$; quan hệ R xác định trên A được cho như sau:

  $R = \{(0,0), (2,1), (0,3), (1,1), (3,0), (1,4), (4,1), (2,2), (2,4), (3,3), (4,4), (1,2), (4,2)\}$

  1) R có phải là một quan hệ tương đương hay không?

  2) Nếu câu 1) là đúng thì chỉ ra phân hoạch của A thành các lớp tương đương theo quan hệ R.

Bài tập 4 (Tập hợp)

- Cho $A = \{0, 1, 2, 3, 4\}$; quan hệ R xác định trên A được cho như sau:

  $R = \{(0,0), (2,1), (0,3), (1,1), (3,0), (1,4), (4,1), (2,2), (2,4), (3,3), (4,4), (1,2), (4,2)\}$

  1) *R* có phải là một quan hệ tương đương hay không?

  2) Nếu câu 1) là đúng thì chỉ ra phân hoạch của *A* thành các lớp tương đương theo quan hệ *R*.

  Lời giải:

  1. R là quan hệ tương đương vì có đủ 3 tính chất: phản xạ, đối xứng và bắc cầu.
  2. Gọi A_i là các tập xác định như sau: $A_i = \{x \in A \mid xRi\}$ với $i \in A$

     $\rightarrow$ Xác định được phân hoạch của A gồm: $A_0 = \{0, 3\}$; $A_1 = \{1, 2, 4\}$

Nội dung Bài 2

1. Lý thuyết tập hợp
2. Logic mệnh đề
3. Logic vị từ
4. Thuật toán và độ phức tạp
5. Bài tập

Một số khái niệm của logic mệnh đề

- Mệnh đề: là một câu khẳng định hoặc đúng hoặc sai.
- Ví dụ:
  - "Hà Nội là thủ đô của Việt Nam" là một mệnh đề đúng.
  - "(5 < 3)" là một mệnh đề sai, "(5 > 3)" là một mệnh đề đúng.
  - "(a<7)" không phải là mệnh đề vì nó không biết khi nào đúng khi nào sai.
- Giá trị chân lý của mệnh đề: mỗi mệnh đề chỉ có một trong 2 giá trị "đúng", ký hiệu là "T", giá trị "sai", ký hiệu là "F". Tập {T, F} được gọi là giá trị chân lý của mệnh đề.
- Ký hiệu: ta ký hiệu mệnh đề bằng các chữ cái in thường (a, b, p, q, r, s, t). *Mỗi* mệnh đề còn được gọi là một công thức. Từ khái niệm về mệnh đề, giá trị chân lý của mỗi mệnh đề, ta xây dựng nên các mệnh đề phức hợp (được gọi là công thức) thông qua các phép toán trên mệnh đề.

Các phép toán của logic mệnh đề [1/2]

Cho p và q là hai mệnh đề

- Phép phủ định: ¬
  - $\neg p$ là mệnh đề, đọc là "Không phải p"
  - Mệnh đề cho giá trị đúng nếu p sai và cho giá trị sai nếu p đúng
- Phép hội: ∧ (AND)
  - $p \wedge q$ là mệnh đề, đọc là "p và q"
  - 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
- Phép tuyển: ∨ (OR)
  - $p \vee q$ là mệnh đề, đọc là "p hoặc q"
  - 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

Các phép toán của logic mệnh đề (2/2)

Cho p và q là hai mệnh đề

- Phép tuyển loại: $\oplus$ (XOR)
  - $p \oplus q$ là mệnh đề đọc là "hoặc p hoặc q"
  - $p \oplus q$ có giá trị T khi một trong p hoặc q có giá trị T, có giá trị F trong các trường hợp khác còn lại
- Phép kéo theo: $\Rightarrow$
  - $p \Rightarrow q$ là mệnh đề đọc là "p kéo theo q; nếu p thì q; p là cần/đủ cho q; q khi p; điều kiện cần cho p là q;…"
  - $p \Rightarrow q$ có giá trị F khi p là T và q là F, có giá trị T trong các trường hợp khác còn lại
  - Ví dụ: Nếu tôi được bầu, tôi sẽ giảm thuế; …
  - Chú ý: if p then *(do…)* trong lập trình khác với if p then q trong logic
- Phép tương đương: $\Leftrightarrow$
  - $p \Leftrightarrow q$ là mệnh đề đọc là "p tương đương q; nếu p thì q và ngược lại; …"
  - 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

Bảng giá trị chân lý

Bảng giá trị chân lý các phép toán mệnh đề							
p	q	$\neg p$	$p \wedge q$ (AND)	$p \vee q$ (OR)	$p \oplus q$ (XOR)	$p \Rightarrow q$	$p \Leftrightarrow q$
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

Bảng các thao tác bit tương ứng				
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

Một số khái niệm

- Thỏa được
  - Một mệnh đề là thỏa được nếu nó đúng với một bộ giá trị chân lý nào đó của các mệnh đề thành phần
- Không thỏa được
  - Một mệnh đề là không thỏa được nếu nó sai với mọi bộ giá trị chân lý của các mệnh đề thành phần
- Vững chắc
  - Một mệnh đề là vững chắc nếu nó đúng với mọi bộ giá trị chân lý của các mệnh đề thành phần

Các mệnh đề tương đương logic (1/2)

- Hai mệnh đề a và b được gọi là tương đương logic nếu chúng có cùng giá trị chân lý với mọi bộ giá trị chân lý của các mệnh đề thành phần
- Ký hiệu: $a \equiv b$
- Một số mệnh đề tương đương cơ bản
  - $a \vee F \equiv a$
  - $a \wedge F \equiv F$
  - $a \vee T \equiv T$
  - $a \wedge T \equiv a$
  - $a \Rightarrow b \equiv \neg a \vee b$
  - $a \Leftrightarrow b \equiv (a \Rightarrow b) \wedge (b \Rightarrow a)$
  - $\neg(\neg a) \equiv a$

Các mệnh đề tương đương logic (2/2)

- Luật giao hoán
  - $a \lor b \equiv b \lor a$
  - $a \land b \equiv b \land a$
- Luật kết hợp
  - $(a \lor b) \lor \mathrm{c} \equiv a \lor (b \lor \mathrm{c})$
  - $(a \land b) \land \mathrm{c} \equiv a \land (b \land \mathrm{c})$
- Luật phân phối
  - $a \land (b \lor \mathrm{c}) \equiv (a \land b) \lor (a \land c)$
  - $a \lor (b \land \mathrm{c}) \equiv (a \lor b) \land (a \lor c)$
- Luật De Morgan
  - $\neg(a \lor b) \equiv \neg a \land \neg b$
  - $\neg(a \land b) \equiv \neg a \lor \neg b$

Dạng chuẩn tắc hội (1/2)

- Một câu - mệnh đề tuyển là tuyển của các mệnh đề nguyên thủy
- Câu tuyển có dạng $p_1 \vee p_2 \vee ... \vee p_n$ trong đó p_i là các mệnh đề nguyên thủy
- Một công thức ở dạng chuẩn tắc hội nếu nó là hội của các câu tuyển
  - $(a \vee e \vee f \vee g) \wedge (b \vee \mathrm{c} \vee d)$

Dạng chuẩn tắc hội (2/2)

- Ta có thể biến đổi một công thức bất kỳ về dạng chuẩn tắc hội bằng cách biến đổi theo các nguyên tắc sau:
  - Khử các phép tương đương: $a \Leftrightarrow b \equiv (a \Rightarrow b) \land (b \Rightarrow a)$
  - Khử các phép kéo theo: $a \Rightarrow b \equiv \neg a \lor b$
  - 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: $a \lor (b \land \mathrm{c}) \equiv (a \lor b) \land (a \lor c)$

Thứ tự ưu tiên của các phép toán logic

- Thường ta sử dụng "(" và ")" để định rõ thứ tự thực hiện
- Để ngắn gọn đôi khi các phép toán được viết liền
- Phép phủ định mức ưu tiên cao nhất: $\neg a \wedge b \equiv \neg(a) \wedge b$
- Phép hội $\wedge$ ưu tiên hơn phép tuyển $\vee$:
  $p \vee q \wedge r \equiv p \vee (q \wedge r)$

Phép toán	Mức ưu tiên
$\neg$	1
$\wedge$ $\vee$	2 3
$\rightarrow$ $\leftrightarrow$	4 5

- Các phép kéo theo $\rightarrow$ và tương đương $\leftrightarrow$ có mức ưu tiên thấp nhất: $p \rightarrow q \vee r \equiv p \rightarrow (q \vee r)$
  Chú ý: cần phải có dấu "(...)" để phân biệt các phép $\rightarrow$ và $\leftrightarrow$

Bài tập 1 (Logic mệnh đề)

- Sử dụng phương pháp bảng chân lý chứng minh sự tương đương logic giữa các mệnh đề:
  - Các mệnh đề tương đương cơ bản
  - Các luật
- Sử dụng các mệnh đề tương đương cơ bản và các luật giao hoán, kết hợp, phân phối, De Morgan chứng minh sự tương đương logic giữa các mệnh đề:
  - $\neg(p \lor (\neg p \land q)) \equiv \neg p \land \neg q$

Bài tập 1 (Logic mệnh đề)

- Ví dụ: Sử dụng phương pháp bảng chân lý chứng minh sự tương đương logic giữa:
  - Mệnh đề tương đương cơ bản: $a \Leftrightarrow b \equiv (a \Rightarrow b) \wedge (b \Rightarrow a)$
  - Luật De Morgan: $\neg(a \vee b) \equiv \neg a \wedge \neg b$

a	b	$a \Leftrightarrow b$	$a \Rightarrow b$	$b \Rightarrow a$	$(a \Rightarrow b) \wedge (b \Rightarrow a)$
T	T	T	T	T	T
T	F	F	F	T	F
F	T	F	T	F	F
F	F	T	T	T	T

a	b	$\neg a$	$\neg b$	$\neg(a \vee b)$	$\neg a \wedge \neg b$
T	T	F	F	F	F
T	F	F	T	F	F
F	T	T	F	F	F
F	F	T	T	T	T

Bài tập 1 (Logic mệnh đề)

- Sử dụng các mệnh đề tương đương cơ bản và các luật giao hoán, kết hợp, phân phối, De Morgan chứng minh sự tương đương logic giữa các mệnh đề:
  - $\neg(p \vee (\neg p \wedge q)) \equiv \neg p \wedge \neg q$
- Biến đổi:

$$\neg(p \vee (\neg p \wedge q)) \underset{\text{De Morgan}}{\equiv} \neg p \wedge \neg(\neg p \wedge q)) \underset{\text{De Morgan}}{\equiv} \neg p \wedge (\neg(\neg p) \vee \neg q)$$

$$\underset{\text{Tương đương cơ bản}}{\equiv} \neg p \wedge (p \vee \neg q) \underset{\text{Phân phối}}{\equiv} (\neg p \wedge p) \vee (\neg p \wedge \neg q) \underset{\text{Tương đương cơ bản}}{\equiv} F \vee (\neg p \wedge \neg q)$$

$$\underset{\text{Tương đương cơ bản}}{\equiv} \neg p \wedge \neg q$$

Bài tập 2 (Logic mệnh đề)

- Chứng minh các mệnh đề sau là vững chắc

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

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

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

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

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

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

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

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

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

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

Bài tập 2 (Logic mệnh đề)

- Ví dụ: Chứng minh mệnh đề
  $S \equiv ((p \vee q) \wedge (p \Rightarrow r) \wedge (q \Rightarrow r)) \Rightarrow r$ là vững chắc.

Đặt $S_1 \equiv ((p \vee q) \wedge (p \Rightarrow r) \wedge (q \Rightarrow r))$

p	q	r	$p \vee q$	$p \Rightarrow r$	$q \Rightarrow r$	S_1	S
T	T	T	T	T	T	T	T
T	T	F	T	F	F	F	T
T	F	T	T	T	T	T	T
T	F	F	T	F	T	F	T
F	T	T	T	T	T	T	T
F	T	F	T	T	F	F	T
F	F	T	F	T	T	F	T
F	F	F	F	T	T	F	T

Bài tập 3 (Logic mệnh đề)

- 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) $\boxed{\neg(p \Leftrightarrow q) \equiv \neg p \Leftrightarrow q}$

p	q	$\neg p$	$p \Leftrightarrow q$	Vế trái	Vế phải
T	T	F	T	F	F
T	F	F	F	T	T
F	T	T	F	T	T
F	F	T	T	F	F

Bài tập 4 (Logic mệnh đề)

- Chuẩn hóa về dạng chuẩn tắc hội
  - $(p \Rightarrow q) \vee \neg(r \vee \neg s)$

- Áp dụng các mệnh đề tương đương cơ bản và các luật ta có:

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

Nội dung Bài 2

1. Lý thuyết tập hợp
2. Logic mệnh đề
3. Logic vị từ
4. Thuật toán và độ phức tạp
5. Bài tập

Logic vị từ

- Mệnh đề: là một câu khẳng định hoặc đúng hoặc sai.
  - Có những khẳng định chưa phải là một mệnh đề, thường liên quan đến các biến.
  - Ví dụ: P(x) = "x > 3" thường thấy trong chương trình máy tính như:

```
if(x > 3){
    …   // Ví dụ câu lệnh C
}
```

- Vị từ P: ứng với phần vị ngữ - vị từ trong khẳng định x>3
  - P(x): một phát biểu ≡ giá trị của hàm mệnh đề P tại x
  - $P(x_0)$: trở thành mệnh đề ứng với x_0 cho trước
- Lượng từ hóa - quantification
  - Cách thức tạo ra mệnh đề từ hàm mệnh đề

Lượng từ hóa - quantification

- Hai kiểu lượng từ hóa quan trọng đặc biệt:
  - Lượng từ hóa với mọi – universal quantification:
    sử dụng lượng từ với mọi – universal quantifier, ký hiệu $\forall$
  - Lượng từ tồn tại – existential quantification:
    sử dụng lượng từ tồn tại – existential quantifier, ký hiệu $\exists$
- Định nghĩa 1: Lượng từ với mọi của P(x) ký hiệu là $\forall x P(x)$ là một mệnh đề "P(x) đúng với mọi x thuộc trường đang xét".
  - Ví dụ: Cho hàm mệnh đề $P(x)$ = "$x^2 + x + 41$ là nguyên tố". Khi đó ta có mệnh đề $\forall x P(x)$ trên miền các số tự nhiên [0..39].
- Định nghĩa 2: Lượng từ tồn tại của hàm mệnh đề P(x) ký hiệu là: $\exists x P(x)$ là một mệnh đề "Tồn tại một phần tử x trong không gian đang xét, tại đó P(x) là đúng".
  - Ví dụ: Cho P(x) là hàm mệnh đề "$x > 3$". Khi đó chúng ta có mệnh đề $\exists x P(x)$ trong không gian các số thực.

Đặc điểm của logic vị từ

- Logic mệnh đề
  - Khả năng biểu diễn giới hạn trong phạm vi thế giới các sự kiện
- Logic vị từ
  - Cho phép mô tả thế giới với các đối tượng, các thuộc tính của đối tượng, các mối quan hệ giữa các đối tượng
  - Đối tượng: một cái bàn, một cái cây, một con người, một cái nhà, một con số, …
  - Tính chất - thuộc tính: cái bàn có bốn chân, làm bằng gỗ, có ngăn kéo,…
  - Quan hệ: cha con, anh em, bạn bè (giữa con người), bên trong, bên ngoài, nằm trên, nằm dưới (giữa các đồ vật),…
  - Hàm: một trường hợp riêng của quan hệ, với mỗi đầu vào ta có một giá trị hàm duy nhất

Cú pháp của logic vị từ (1/4)

- Các ký hiệu
  - Các ký hiệu hằng: a, b, c, An, Ba, John, …
  - Các ký hiệu biến: x, y, z, u, v, w, …
  - Các ký hiệu vị từ: P, Q, R, S, Like, Friend, …
    - Mỗi vị từ là vị từ của n biến ($n \geq 0$)
    - Vị từ không biến là các ký hiệu mệnh đề
  - Các ký hiệu hàm: f, g, cos, sin, mother, husband, …
    - Mỗi hàm là hàm của n biến ($n \geq 0$)
  - Các ký hiệu kết nối logic: $\wedge$ (hội), $\vee$ (tuyển), $\neg$ (phủ định), $\Rightarrow$ (kéo theo), $\Leftrightarrow$ (kéo theo nhau)
  - Các ký hiệu lượng tử: $\forall$ (mọi), $\exists$ (tồn tại)
  - Các ký hiệu ngăn cách: dấu phẩy, mở ngoặc, đóng ngoặc

Cú pháp của logic vị từ (2/4)

- Các hạng thức - term
  - Để ký hiệu các đối tượng (toán), được xác định đệ quy như sau
    - Các ký hiệu hằng và các ký hiệu biến là hạng thức
    - Nếu $t_1, t_2, \ldots, t_n$ là n hạng thức, và f là một ký hiệu hàm n biến thì $f(t_1, t_2, \ldots, t_n)$ là hạng thức
  - Một hạng thức không chứa biến được gọi là một hạng thức cụ thể - ground term
  - Hai hạng thức bằng nhau nếu cùng tương ứng với một đối tượng
    - Father(John) = Mike
- Công thức nguyên tử - câu đơn
  - Biểu diễn tính chất của đối tượng, hoặc quan hệ giữa các đối tượng, được xác định đệ quy như sau
    - Các ký hiệu vị từ không biến - mệnh đề là công thức nguyên tử
    - Nếu $t_1, t_2, \ldots, t_n$ là n hạng thức, và P là vị từ của n biến thì $P(t_1, t_2, \ldots, t_n)$ công thức nguyên tử

Cú pháp của logic vị từ (3/4)

- Công thức
  - Được xây dựng từ công thức nguyên tử, sử dụng các kết nối logic và các lượng tử, xác định theo đệ quy như sau
    - Các công thức nguyên tử là công thức
    - Nếu G và H là các công thức, thì các biểu thức sau là công thức $(G \wedge H), (G \vee H), (\neg G), (G \Rightarrow H), (G \Leftrightarrow H)$
    - Nếu G là công thức và x là biến thì các biểu thức sau là công thức $\forall x G$, $(\exists x G)$
- Một số quy ước
  - Các công thức không phải công thức nguyên tử gọi là công thức phức hợp (câu phức hợp)
  - Công thức không chứa biến gọi là công thức cụ thể
  - Khi viết công thức ta bỏ đi các dấu ngoặc không cần thiết

Cú pháp của logic vị từ (4/4)

- Lượng tử phổ dụng (hay Lượng từ với mọi) - $\forall$
  - Mô tả tính chất của cả một lớp các đối tượng, mà không cần liệt kê các đối tượng ra
  - $\forall x(Elephant(x) \Rightarrow Color(x, Gray))$
- Lượng từ tồn tại - $\exists$
  - Cho phép tạo ra câu nói đến một đối tượng nào đó trong một lớp đối tượng, có tính chất hoặc thỏa mãn một quan hệ nào đó
  - $\exists x(Student(x) \wedge Inside(x, P301))$
- Literal
  - Là công thức nguyên tử hoặc phủ định của công thức nguyên tử
  - $Play(x, Football), \neg Like(Lan, Rose)$
- Câu tuyển
  - Là tuyển của các literal
  - $Male(x) \vee \neg Like(x, Football)$

Ngữ nghĩa của logic vị từ (1/3)

- Minh họa
  - Là một cách gán cho các biến đối tượng một đối tượng cụ thể, gán cho các ký hiệu hàm một hàm cụ thể, và các ký hiệu vị từ một vị từ cụ thể
  - Ý nghĩa của công thức trong một thế giới hiện thực nào đó
- Ngữ nghĩa của câu đơn
  - Trong một minh họa, mỗi câu đơn sẽ chỉ định một sự kiện cụ thể, có thể đúng (True) hoặc sai (False)
  - $Student(Lan)$
- Ngữ nghĩa của câu phức
  - Được xác định dựa trên ngữ nghĩa của các câu đơn và các kết nối logic
  - $Student(Lan) \land Like(An, Rose)$
  - $Like(An, Rose) \lor \neg Like(An, Tulip)$

Ngữ nghĩa của logic vị từ (2/3)

- Ngữ nghĩa của câu chứa lượng tử
  - Công thức $\forall x G$ là đúng nếu và chỉ nếu mọi công thức nhận được từ G bằng cách thay x bởi một đối tượng trong miền đối tượng đều đúng
    - Ví dụ: Miền đối tượng {An, Ba, Lan}, ngữ nghĩa của câu $\forall x Student(x)$ được xác định là ngữ nghĩa của câu $Student(An) \wedge Student(Ba) \wedge Student(Lan)$
  - Công thức $\exists x G$ là đúng nếu và chỉ nếu một trong các công thức nhận được từ G bằng cách thay x bởi một đối tượng trong miền đối tượng đều đúng
    - Ví dụ: ngữ nghĩa của câu $\exists x Student(x)$ được xác định là ngữ nghĩa của câu $Student(An) \vee Student(Ba) \vee Student(Lan)$
- Các khái niệm công thức thỏa được, không thỏa được, vững chắc, mô hình tương tự logic mệnh đề.

Ngữ nghĩa của logic vị từ (3/3)

- Các lượng tử lồng nhau
  - Có thể sử dụng đồng thời nhiều lượng tử trong câu phức hợp
    - $\forall x \forall y Sibling(x, y) \Rightarrow Relationship(x, y)$
    - $\forall x \exists y Love(x, y)$
- Nhiều lượng tử cùng loại có thể được viết gọn bằng một ký hiệu lượng tử
    - $\forall x, y Sibling(x, y) \Rightarrow Relationship(x, y)$
- Không được phép thay đổi các lượng tử khác loại trong câu
    - $\forall x \exists y Love(x, y)$ Mọi người đều có ai đó yêu

$\neq$

    - $\exists y \forall x Love(x, y)$ Có ai đó mà tất cả mọi người đều yêu

Bài tập 1 (Logic vị từ)

- Chuyển các câu sau sang logic vị từ cấp 1
  1. An không cao
  2. An không cao nhưng bố An cao
  3. An và Ba là anh em
  4. Tất cả nhà nông đều thích mặt trời
  5. Mọi cây nấm đỏ đều không có độc
  6. Có học sinh thích hoa hồng

Bài tập 1 (Logic vị từ)

- Chuyển các câu sau sang logic vị từ cấp 1: Gợi ý
  1. An không cao: Xây dựng vị từ Cao(x) khi đó ta có $\neg Cao(An)$
  2. An không cao nhưng bố An cao: $\neg Cao(An) \wedge Cao(Bo(An))$
  3. An và Ba là anh em: Xây dựng vị từ "là anh em" hay hàm vị từ Bother(x, y) khi đó ta có: Brother(An, Ba)
  4. Tất cả nhà nông đều thích mặt trời: Xây dựng các vị từ "là nhà nông" hay hàm vị từ Farmer(x) và "thích mặt trời" - LoveSun(x), khi đó ta có: $\forall x Farmer(x) \Rightarrow LoveSun(x)$
  5. Mọi cây nấm đỏ đều không có độc: Xây dựng các vị từ "là cây nấm" hay hàm vị từ Mushroom(x), "có màu đỏ" – RedColor(x) và "có độc" - Poisonous(x), khi đó ta có: $\forall x(Mushroom(x) \wedge RedColor(x)) \Rightarrow \neg Poisonous(x)$
  6. Có học sinh thích hoa hồng: Xây dựng vị từ "thích hoa hồng" LoveRose(x) và "là sinh viên" Student(x): $\exists x LoveRose(x) \wedge Student(x)$

Nội dung Bài 2

1. Lý thuyết tập hợp
2. Logic mệnh đề
3. Logic vị từ
4. Thuật toán và độ phức tạp
5. Bài tập

Khái niệm thuật toán

- Thuật toán hoặc giải thuật - algorithm
  - Là một thủ tục giải quyết một vấn đề nào đó trong một số hữu hạn bước
  - Là một tập hữu hạn các chỉ thị được định nghĩa rõ ràng để giải quyết một vấn đề nào đó
  - Là một tập các quy tắc định nghĩa chính xác một dãy các hành động
- Mô tả thuật toán
  - Sử dụng ngôn ngữ tự nhiên
  - Sử dụng dạng giả mã (Pseudo-code)
  - Sử dụng ngôn ngữ lập trình

Ví dụ thuật toán (1/2)

- Thuật toán: Tìm số nguyên lớn nhất trong danh sách n số nguyên (chưa sắp xếp)
- Mô tả thuật toán sử dụng ngôn ngữ tự nhiên:
  - Nếu không có số nào trong danh sách thì không có số lớn nhất
  - Giải sử số đầu tiên là số lớn nhất trong danh sách
  - Với mỗi số còn lại trong danh sách, nếu số này lớn hơn số lớn nhất hiện tại thì coi số này là số lớn nhất trong danh sách
  - Khi tất cả các số trong danh sách đều đã được xem xét, số lớn nhất hiện tại là số lớn nhất trong danh sách
  - Đặc điểm của mô tả kiểu này: **Dài dòng, ít được sử dụng**

Ví dụ thuật toán (2/2)

- Thuật toán: Tìm số nguyên lớn nhất trong danh sách n số nguyên (chưa sắp xếp)
- Mô tả thuật toán sử dụng dạng giả mã:

```
Algorithm: LargestNumber
Input: A list L of numbers.
Output: The largest number in the list L.
largest ← null
for each item in L do
    if item > largest then
        largest ← item
return largest
```

- Đặc điểm: Dễ hiểu, hay được sử dụng, không phụ thuộc vào ngôn ngữ lập trình

Độ phức tạp thuật toán

- Hầu hết các thuật toán được thiết kế làm việc với kích thước dữ liệu đầu vào tùy ý.
- Ví dụ thuật toán ở trên, kích thước dữ liệu đầu vào là số phần tử trong danh sách - n
- Độ phức tạp thời gian - time complexity
  - Xác định lượng thời gian cần thiết để thực hiện giải thuật
  - Được tính là số phép toán cơ bản thực hiện giải thuật
- Độ phức tạp không gian - space complexity
  - Xác định lượng bộ nhớ cần thiết để thực hiện giải thuật
  - Lượng bộ nhớ lớn nhất cần thiết để lưu các đối tượng của thuật toán tại một thời điểm thực hiện thuật toán
- Độ phức tạp của thuật toán thường được biểu diễn như là một hàm của kích thước dữ liệu đầu vào

Khái niệm O-lớn

- $f(n) = O(g(n))$, với n đủ lớn, $f(n)$ không quá một hằng số cố định nhân với $g(n)$
  - n là một số nguyên dương kí hiệu kích thước của dữ liệu đầu vào

$$f(n) = O(g(n)) \Leftrightarrow \exists c, n_0 \text{ sao cho } \forall n \geq n_0, f(n) \leq c \times g(n)$$

- Định lý
  - Nếu $lim_{n \to \infty} [f(n) / g(n)]$ tồn tại và hữu hạn, thì $f(n) = O(g(n))$

Ví dụ về độ phức tạp thuật toán

- Thuật toán: Tìm số nguyên lớn nhất trong danh sách n số nguyên

```
Algorithm: LargestNumber
Input: A list L of numbers.
Output: The largest number in the list L.
largest ← null
for each item in L do
    if item > largest then
        largest ← item
return largest
```

- Độ phức tạp thời gian và không gian: **đều là $O(n)$**

Bài tập 1 (Thuật toán và độ phức tạp của thuật toán)

- Tìm độ phức tạp tính toán của thuật toán sắp xếp nổi bọt (bubble sort):

```
function bubble_sort(List L, number n) // n chiều dài L
for i from n down to 2
      for j from 1 to (i - 1)
            if L[j] > L[j + 1] then //nếu chúng không đúng thứ tự
                  swap(L[j], L[j + 1]) //đổi chỗ chúng cho nhau
            end if
      end for
end for
end function
```

Gợi ý

Độ phức tạp thời gian: $O(n^2)$

Độ phức tạp không gian: $O(n)$

Bài tập 2 (Thuật toán và độ phức tạp của thuật toán)

- Tìm độ phức tạp tính toán của thuật toán đệ quy tìm kiếm nhị phân trên danh sách đã sắp xếp:

```
function binary_search_recursive (A, x, L, R)
if L > R then
    return Fail
else
    i ← (L + R)/2
    if A[i] == x then
        return i
    else if A[i] > x then
            return binary_search(A, x, L, i − 1)
        else
            return binary_search(A, x, i + 1, R)
        end if
    end if
end function
```

Gợi ý

Độ phức tạp thời gian: O(log(n))

Độ phức tạp không gian: O(nlog(n))

Thuật toán tìm kiếm nhị phân không đệ quy

function binary_search_non_recursive (A, x)

i ← 1 //i cận trái miền tìm kiếm

j ← n //j cận phải miền tìm kiếm

while i < j

    m ← ⌊(i + j)/2⌋

    if x > A[m] then i ← m + 1
    else j ← m

    if x = A[i] then vi_tri ← i
    else vi_tri ← 0

//vi_tri: là chỉ số i của số hạng A[i] nếu tìm thấy, 0 nếu không tìm thấy

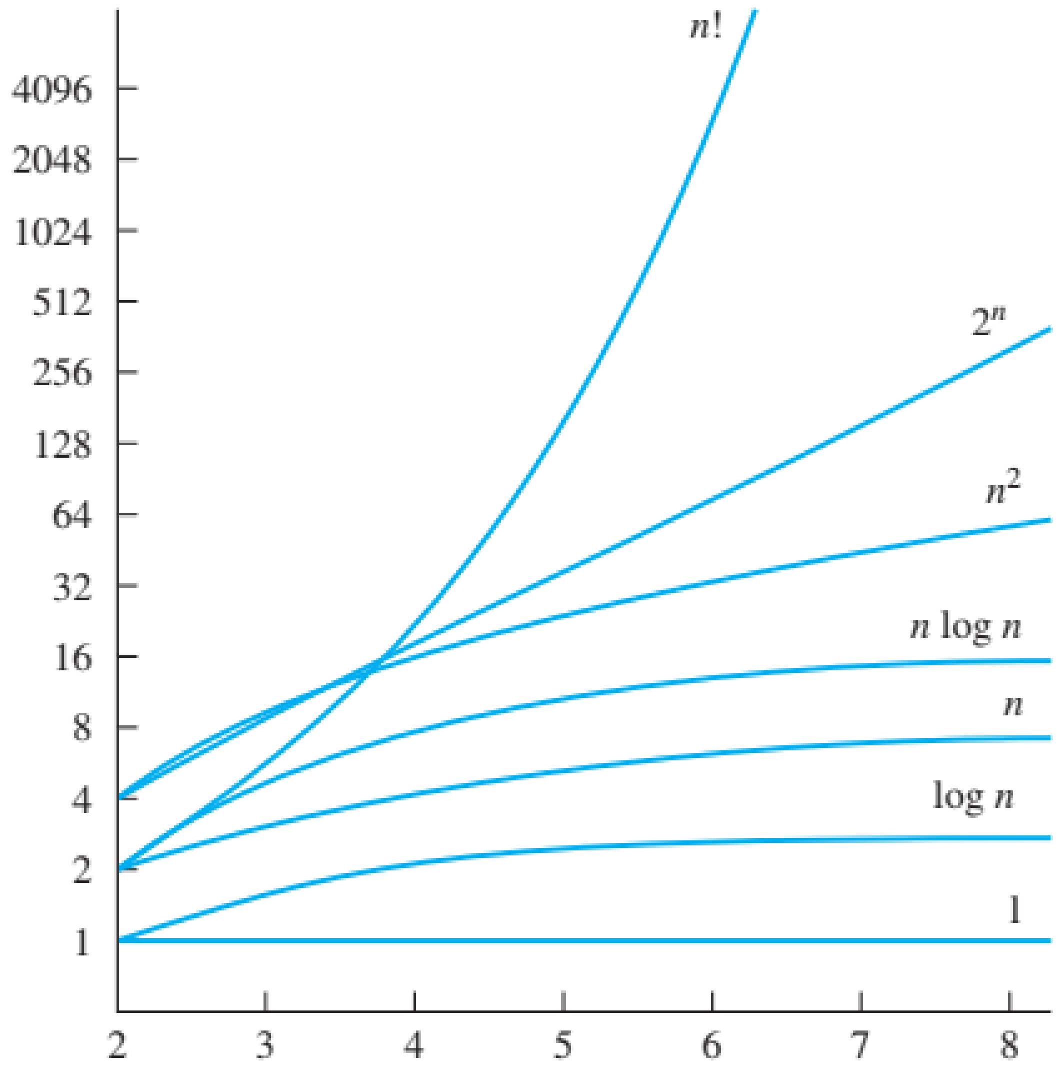
return vi_tri

Gợi ý

Độ phức tạp thời gian: O(log(n))

Độ phức tạp không gian: O(n)

So sánh giá trị theo n của một số hàm cơ bản



Kết thúc Bài 2

- Câu hỏi và thảo luận