

CHƯƠNG 1: KHOA HỌC MÁY TÍNH

Mục lục

- ❑ Nguyên lý cơ bản về thông tin.
- ❑ Thông tin và Logic.
- ❑ Cấu trúc dữ liệu.
- ❑ Giải thuật.

1.1 Nguyên lý cơ bản về thông tin

- ❑ Chuyển đổi cơ số
- ❑ Biểu diễn số
- ❑ Biểu diễn dữ liệu không phải số
- ❑ Các phép toán và độ chính xác
- ❑ QUIZ

1.1.1 Chuyển đổi cơ số

- ❑ Ba hệ cơ số được sử dụng : 2, 10, 16.
- ❑ Thông tin đều được lưu trữ dưới dạng nhị phân
- ❑ Hệ cơ số Hexa để dễ biểu diễn và theo dõi

Decimal	Binary	Hexadecimal	Decimal	Binary	Hexadecimal
0	0000	0	8	1000	8
1	0001	1	9	1001	9
2	0010	2	10	1010	A
3	0011	3	11	1011	B
4	0100	4	12	1100	C
5	0101	5	13	1101	D
6	0110	6	14	1110	E
7	0111	7	15	1111	F
			16	10000	10

1.1.1 Chuyển đổi cơ số

□ Các chuyển đổi thường gặp

□ Chuyển đổi từ Hexa, nhị phân sang hệ thập phân :

$$\begin{aligned}(12A)_{16} &= 1 \times 16^2 + 2 \times 16^1 + A \times 16^0 \\ &= 256 + 32 + 10 \\ &= (298)_{10}\end{aligned}$$

$$\begin{aligned}(1100100)_2 &= 1 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 \\ &= 64 + 32 + 4 \\ &= (100)_{10}\end{aligned}$$

$$\begin{aligned}(0.4B)_{16} &= 4 \times 16^{-1} + B \times 16^{-2} \\ &= 4 / 16 + 11 / 16^2 \\ &= 0.25 + 0.04296875 \\ &= (0.29296875)_{10}\end{aligned}$$

$$\begin{aligned}(0.01011)_2 &= 0 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} + 1 \times 2^{-5} \\ &= 0.25 + 0.0625 + 0.03125 \\ &= (0.34375)_{10}\end{aligned}$$

1.1.1 Chuyển đổi cơ số

- ❑ Chuyển đổi từ hệ 10 sang hệ nhị phân
 - ❑ Chuyển đổi phần nguyên

Phần dư

$2 \overline{) 59} \cdots 1$	\rightarrow	(1) $59 / 2 = 29$ phần dư 1	
$2 \overline{) 29} \cdots 1$	\rightarrow	(2) $29 / 2 = 14$ phần dư 1	
$2 \overline{) 14} \cdots 0$	\rightarrow	(3) $14 / 2 = 7$ phần dư 0	
$2 \overline{) 7} \cdots 1$	\rightarrow	(4) $7 / 2 = 3$ phần dư 1	
$2 \overline{) 3} \cdots 1$	\rightarrow	(5) $3 / 2 = 1$ phần dư 1	
$2 \overline{) 1} \cdots 1$	\rightarrow	(6) $1 / 2 = 0$ phần dư 1	

$\rightarrow (59)_{10} = (111011)_2$

- ❑ Chuyển đổi phần thập phân

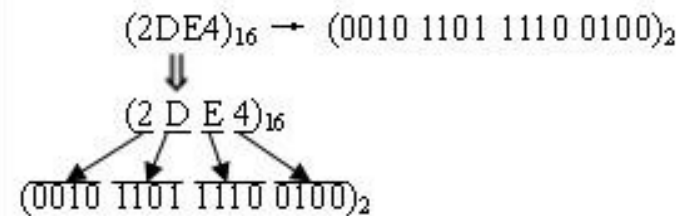
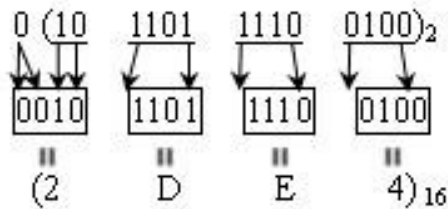
$\rightarrow (0.59375)_{10} = (0.10011)_2$

$0.59375 \times 2 =$	1	$.1875$	\rightarrow (1)	
$0.1875 \leftarrow \times 2 =$	0	$.375$	\rightarrow (2)	
$0.375 \leftarrow \times 2 =$	0	$.75$	\rightarrow (3)	
$0.75 \leftarrow \times 2 =$	1	$.5$	\rightarrow (4)	
$0.5 \leftarrow \times 2 =$	1	$.0$	\leftarrow	

1.1.1 Chuyển đổi cơ số

- Chuyển đổi giữa hệ 16 sang hệ nhị phân

$$(10110111100100)_2 \square (10\ 1101\ 1110\ 0100)_2 \square (2DE4)_{16}$$



1.1.2 Các dạng biểu diễn số học

Số thập phân	Thập phân dạng vùng	Tương thích cao với dữ liệu văn bản (cũng được biết tới như số thập phân gói mở)
	Thập phân gói đóng	Tốc độ xử lý nhanh hơn
Số nhị phân	Dấu phẩy tĩnh	Sử dụng cho dữ liệu số nguyên, ví dụ như chỉ số mảng...
	Dấu phẩy động	Sử dụng cho dữ liệu số thực như trong tính toán khoa học...

- ❑ Biểu diễn số nguyên theo BCD (Binary coded Decimal Codes):
 - ❑ Dùng 4 bit để mã hóa cho các chữ số thập phân từ 0 đến 9.
 - ❑ còn 6 tổ hợp không sử dụng: 1010, 1011, 1100, 1101, 1110, 1111

1.1.2 Các dạng biểu diễn số học

- Số thập phân khoanh vùng:

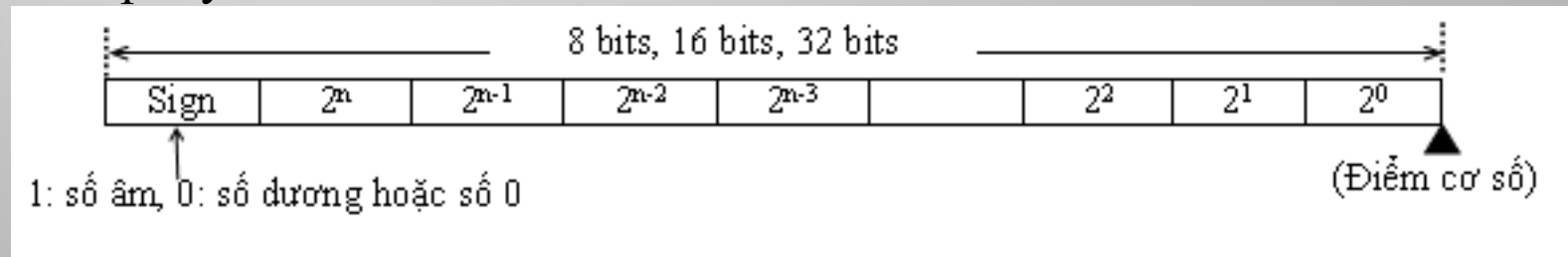
		1		2		3	+	4	
+1234	0011	0001	0011	0010	0011	0011	1100	0100	Sign bits 1100: Positive or zero
		1		2		3	-	4	
-1234	0011	0001	0011	0010	0011	0011	1100	0100	1101: Negative

Zone bits	Numeral bits	Sign bits	Numeral bits	Zone bits: 0011
-----------	--------------	-----------	--------------	-----------------

- Số thập phân đóng gói:

	0	1	2	3	4	+		0	1	2	3	4	-
+1234	0000	0001	0010	0011	0100	1100	-1234	0000	0001	0010	0011	0100	1101

- Dấu phẩy tĩnh:



1.1.2 Các dạng biểu diễn số học

- Dấu phẩy động: dùng để biểu diễn số thực âm và dương
 - Đối với dạng 32 bit.

1 bit	8 bits		23 bits
0	10000100		110100000000000000000000
□	□	□	□
Dấu phân định trị ±	Phân mũ e	Điểm cơ số	Phân định trị m

- Đây là chuẩn do IEEE754 đề xướng.

Công thức xác định giá trị của số thực:

$$X = (-1)^S * 1.m * 2^{e-127}$$

1.3.1 Các biểu diễn phi số

- ❑ Đây là loại biểu diễn cho các dữ liệu không phải là con số.
- ❑ Có thể là các dữ liệu âm thanh, ảnh tĩnh và ảnh động.
- ❑ Biểu diễn ký tự
- ❑ Dùng một trong các bảng mã sau

Tên mã	Giải thích
EBCDIC	Mã được xây dựng bởi IBM, trong đó 8 bit biểu diễn một ký tự.
ASCII	Mã 7-bit được xây dựng bởi ANSI (American National Standards Institute) dùng cho các PC.
Mã ISO	Mã ISO: là mã ký tự do Tổ chức tiêu chuẩn quốc tế (ISO) quy định dưới dạng quy tắc "ISO 646", dựa trên ASCII. Dùng 7 bit để trao đổi thông tin.
Unicode	Mã UNICODE: là loại sử dụng tất cả các loại ký tự trên thế giới, trong đó bao gồm cả chữ Hán, bằng một tiêu chuẩn mã hóa 2 byte thống nhất.
EUC	bảng mã 2-byte hoặc một-byte được dùng cho tiếng Nhật và tiếng Hàn Quốc (và một phần nhỏ tiếng Trung Hoa) hoạt động trên Unix.

1.3.1 Các biểu diễn phi số

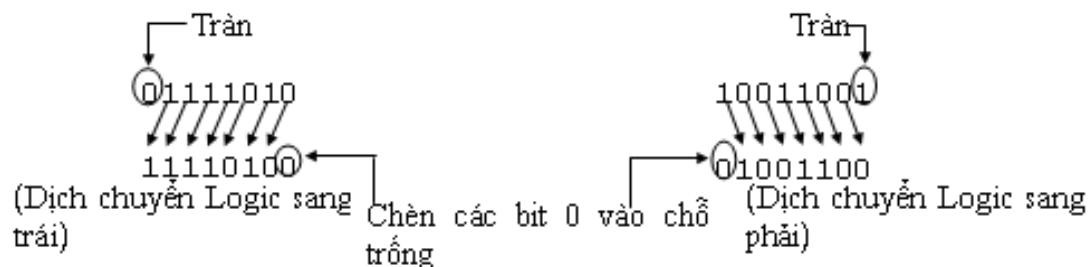
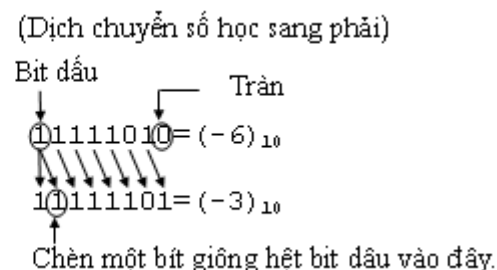
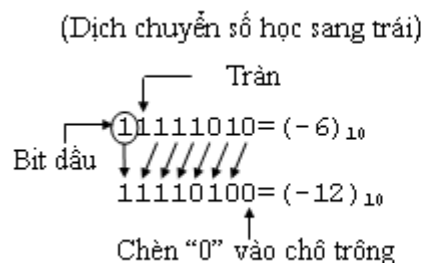
❑ Biểu diễn âm thanh và hình ảnh*:

Ảnh tĩnh	GIF	Một định dạng lưu ảnh, hiển thị được 256 màu
	JPEG	Là dạng thức lưu trữ nén ảnh tĩnh dưới dạng đủ mẫu do ISO/IEC và ITU-T cùng quy định.
Ảnh động	MPEG	MPEG: là dạng thức lưu ảnh động dưới dạng đủ mẫu, không dùng để lưu giữ ảnh tĩnh. Do ISO/IEC quy định.
	MPEG-1	Có thể nén tín hiệu video tới 1.5Mbit/s với chất lượng VHS và âm thanh lập thể (stereo audio) với tốc độ 192 bit/s. Nó được dùng để lưu trữ video và âm thanh trên CD-ROM.
	MPEG-2	MPEG-2 có khả năng mã hoá tín hiệu truyền hình ở tốc độ 3-15Mbit/s và truyền hình độ nét cao ở tốc độ tới 15-30Mbit/s. Dùng để lưu trữ các video hoặc các ảnh thời gian thực
	MPEG-4	Chuẩn lưu trữ dành cho các thiết bị di động
Âm thanh	PCM	(Pulse Code Modulation) Điều biến mã xung biến tín hiệu tương tự thành tín hiệu số.
	MIDI	(Musical Instrument Digital Interface - Giao diện kỹ thuật số dành cho nhạc cụ) là chuẩn công nghiệp về nghi thức giao thông điện tử định rõ các nốt âm nhạc trong nhạc cụ điện tử như là bộ tổng hợp chính xác và ngắn gọn, để nhạc cụ điện tử và máy tính trao đổi dữ liệu.

1.1.4 Các phép tính và độ chính xác

❑ Phép dịch bit

	Dịch chuyển số học	Dịch chuyển Logic
Sang trái	Dịch chuyển số học sang trái	Dịch chuyển Logic sang trái.
Sang phải	Dịch chuyển số học sang phải.	Dịch chuyển Logic sang phải.



1.1.4 Các phép tính và độ chính xác

❑ Các lỗi:

- ❑ Xuất hiện do có sự giới hạn trong sức chứa của các thanh ghi, dẫn đến có sai số giữa kết quả tính và giá trị thật.
- ❑ Rounding error: Sai số làm tròn, lỗi xuất hiện khi 1 kết quả thập phân được làm tròn lên hoặc xuống.
- ❑ Tránh việc cộng/trừ các giá trị quá chênh lệch bằng cách tính các giá trị nhỏ nhất trước, sẽ bớt ảnh hưởng của sai số làm tròn.

$$\begin{array}{r}) 356.3622 \\ - 356.3579 \\ \hline 0.0043 \end{array}$$

↑
Kết quả quá nhỏ, coi như bằng 0.

$$\begin{array}{r} 356.3622 \\ -) 0.000015 \\ \hline 356.3622 \end{array} \quad \leftarrow \text{Số bị trừ quá nhỏ, coi như không thay đổi giá trị}$$

QUIZ

Q1: Chuyển đổi 100.5 sang hệ nhị phân, bát phân, thập lục phân.

Q2: Thực hiện phép dịch chuyển số học và dịch chuyển logic sang phải 3 bit trên số nhị phân 8 bit 11001100.

1.2 Thông tin và logic

- ❑ Các phép toán logic
- ❑ BNF
- ❑ Kí pháp Ba Lan ngược
- ❑ QUIZ

1.2.1 Các phép toán logic

- ❑ Và (AND)
- ❑ Hoặc (OR)
- ❑ Phủ định (NOT)
- ❑ Tổng loại trừ (XOR).
- ❑ Định luật DeMorgan

$$\overline{(A \cdot B)} = \bar{A} + \bar{B}$$

$$\overline{(A + B)} = \bar{A} \cdot \bar{B}$$

A	B	Tích logic	Tổng logic	Tổng loại trừ logic	Phủ định logic	
		$A \cdot B$	$A + B$	$A \oplus B$	\bar{A}	\bar{B}
0	0	0	0	0	1	1
0	1	0	1	1	1	0
1	0	0	1	1	0	1
1	1	1	1	0	0	0

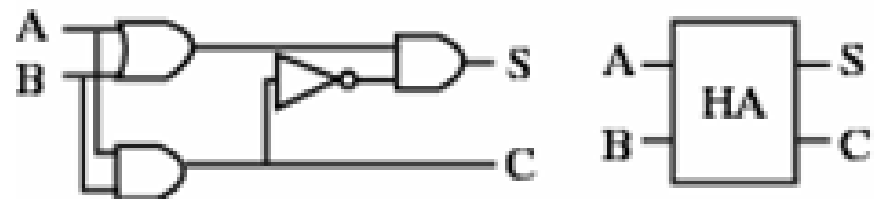
1.2.1 Các phép toán logic

- ❑ Mạch thực hiện phép cộng số nhị phân 1 bit
- ❑ Bao gồm các mạch logic: AND, OR, NOT.
- ❑ Hai loại bộ cộng:
 - Bộ bán cộng, không đưa vào tổng số nhớ từ bit thấp hơn.
 - Bộ cộng đầy đủ, đưa vào tổng số nhớ từ các bit thấp hơn.
- ❑ Bộ cộng bán cộng:
 - C biểu thị cờ nhớ và S là bit thấp của kết quả phép cộng.
 - C là tích logic và S là tổng loại trừ .

$$C = A \text{ and } B$$

$$S = A \text{ xor } B$$

A		B		C	S
0	+	0	=	0	0
0	+	1	=	0	1
1	+	0	=	0	1
1	+	1	=	1	0



Kí hiệu đơn giản²¹

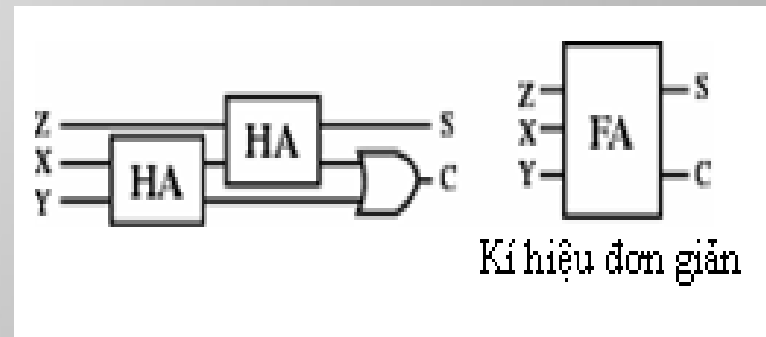
1.2.1 Các phép toán logic

□ Bộ cộng đầy đủ:

- Một bộ cộng đầy đủ có 3 đầu vào, 1 trong số đó là cờ nhớ từ bit thấp hơn.
- Bộ cộng đầy đủ gồm 2 bộ bán cộng kết hợp với nhau
- $S = X \text{ xor } Y \text{ xor } Z$

$$C = (X \text{ and } Y) \text{ or } (Z \text{ and } (X \text{ xor } Y))$$

X		Y		Z		C	S
0	+	0		0	=	0	0
0	+	0		1	=	0	1
0	+	1		0	=	0	1
0		1		1	=	1	0
1		0		0	=	0	1
1		0		1	=	1	0
1		1		0	=	1	0
1	+	1		1	=	1	1



1.2.2 BNF - Backus-Naur Form

- ❑ Được dùng để định nghĩa văn phạm của 1 ngôn ngữ lập trình (định nghĩa cú pháp).
- ❑ Dạng cơ bản của BNF:
- ❑ Biểu thức BNF chứa phép tuần tự, lặp và chọn.

Tuần tự: $\langle x \rangle ::= \langle a \rangle \langle b \rangle$

Lặp $\langle x \rangle ::= \langle a \rangle \dots$ “phần tử cú pháp x là 1 dãy lặp của kí tự a”. Nó cũng có nghĩa là kí tự a được lặp 1 hoặc nhiều lần.

Chọn:

$\langle x \rangle ::= \langle a \mid b \rangle$ “phần tử cú pháp x nhận kí tự a hoặc kí tự b”. Nếu 1 trong các lựa chọn bị bỏ qua, biểu thức tiếp theo được sử dụng:

$\langle x \rangle ::= [\langle a \rangle]$ “phần tử cú pháp x nhận kí tự a hoặc kí tự rỗng”. Kí hiệu “[]” có nghĩa có thể bỏ qua

1.2.2 BNF - Backus-Naur Form

- Kí hiệu kết thúc và không kết thúc:
 - + Một phần tử cú pháp đã được định nghĩa có thể dùng để định nghĩa phần tử khác hoặc chính nó : “**kí hiệu không kết thúc**”.
 - + Các ký tự còn lại : “**kí hiệu kết thúc**”.

Trong định nghĩa sau, “<x>” là kí hiệu không kết thúc trong khi a, b và c là kí hiệu kết thúc.

$\langle y \rangle ::= \langle a \rangle \langle x \rangle$

$\langle x \rangle ::= \langle b \rangle \langle c \rangle$

Ví dụ: quy tắc cú pháp của “hàng dấu phẩy động”:

$\langle \text{hàng dấu phẩy động} \rangle ::= [\langle \text{dấu} \rangle \langle \text{hàng cơ số} \rangle [\langle \text{số mũ} \rangle] |$
 $[\langle \text{dấu} \rangle \langle \text{chuỗi số} \rangle \langle \text{số mũ} \rangle]$

$\langle \text{hàng cơ số} \rangle ::= [\langle \text{chuỗi số} \rangle \langle . \rangle \langle \text{chuỗi số} \rangle | \langle \text{chuỗi số} \rangle \langle . \rangle]$

$\langle \text{số mũ} \rangle ::= \langle E \rangle [\langle \text{dấu} \rangle \langle \text{chuỗi số} \rangle]$

$\langle \text{chuỗi số} \rangle ::= \langle \text{chữ số} \rangle | \langle \text{chuỗi số} \rangle \langle \text{chữ số} \rangle$

$\langle \text{chữ số} \rangle ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$

$\langle \text{dấu} \rangle ::= + | -$

1.2.3 Kí pháp Ba Lan ngược

- ❑ Phương pháp biểu diễn các công thức toán học sang dạng biểu diễn dễ xử lý hơn bởi máy tính.
- ❑ Toán tử được đặt vào cuối thay vì đặt vào giữa của công thức.
- ❑ Ví dụ, $X = A + B * C$ nghĩa là “tính tích của B và C, cộng A, sau đó đưa kết quả tới X”, được biểu diễn như sau:

$XABC*+=$

QUIZ

Q1: Chuyển công thức “ $(a + b) \times (c - d)$ ” sang dạng kí pháp Ba Lan ngược.

1.3 Cấu trúc dữ liệu

- ❑ Mảng
- ❑ Danh sách
- ❑ Ngăn xếp và hàng đợi
- ❑ Cây
- ❑ Băm
- ❑ QUIZ

1.3.1 Mảng

- ❑ Mảng là một cấu trúc dữ liệu tạo bởi nhiều dữ liệu cùng kiểu xếp liên nhau trong bộ nhớ.

“Mảng một chiều” là khái niệm như sau:

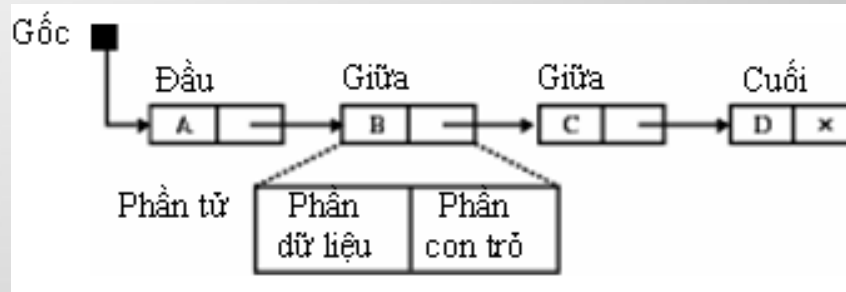
Chỉ số	1	2	3	4	...	25	26
Mảng T	a	b	c	d	...	y	z

Một **mảng hai chiều** là khái niệm như sau

	Cột 1	Cột 2	Cột 3	Cột 4
Dòng 1	$a(1,1)$	$a(1,2)$	$a(1,3)$	$a(1,4)$
Dòng 2	$a(2,1)$	$a(2,2)$	$a(2,3)$	$a(2,4)$
Dòng 3	$a(3,1)$	$a(3,2)$	$a(3,3)$	$a(3,4)$

1.3.2 Danh sách

- ❑ Tập của các dữ liệu giống nhau hoặc tương tự nhau được đặt một cách logic trên một hàng (tuyến tính);
- ❑ Các phần tử của danh sách có thể đặt ở những vị trí độc lập và được liên kết bởi các con trỏ.
- ❑ Cấu trúc của danh sách:

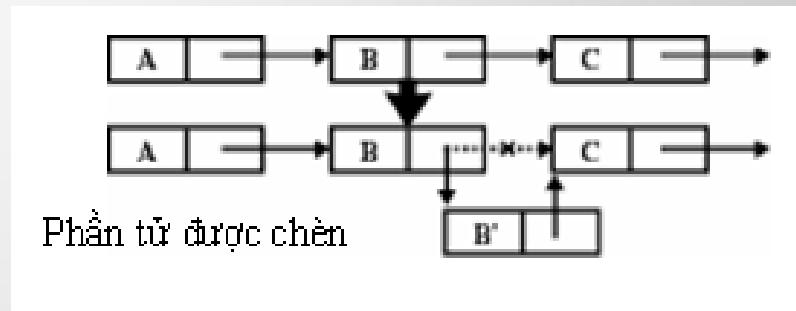


Danh sách liên kết một chiều

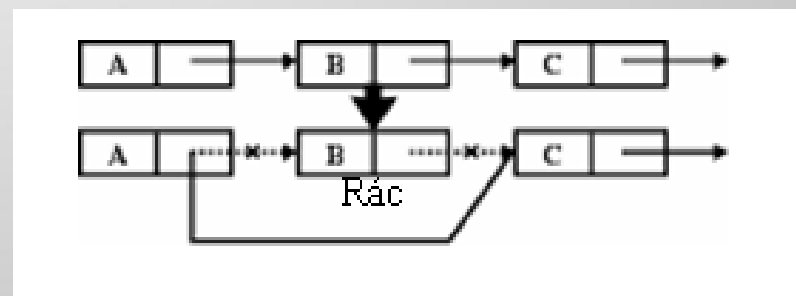
1.3.2 Danh sách

- Các thao tác với danh sách:

+ **Chèn**

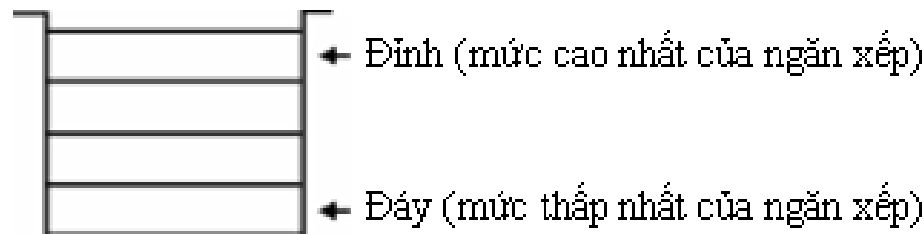


+ **Xóa**

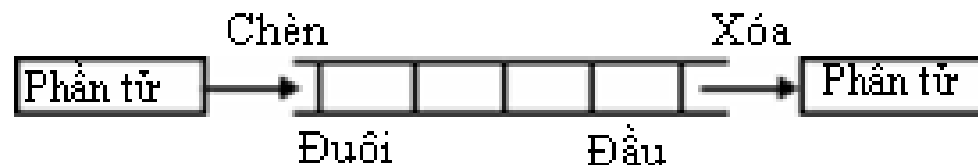


1.3.3 Ngăn xếp và hàng đợi

Ngăn xếp là một cấu trúc dữ liệu trong đó dữ liệu chèn thêm vào và xóa đi đều nằm ở cùng một đầu của danh sách. Nó được minh họa như hình dưới.

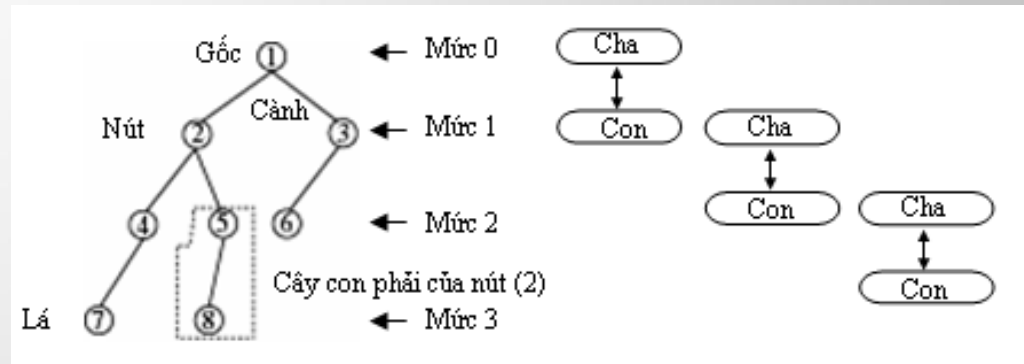


Hàng đợi là một cấu trúc dữ liệu mà thao tác chèn được thực hiện ở một đầu và thao tác xóa (lấy ra) được thực hiện ở đầu kia. Dựa trên khái niệm, nó có được mô tả như sau.

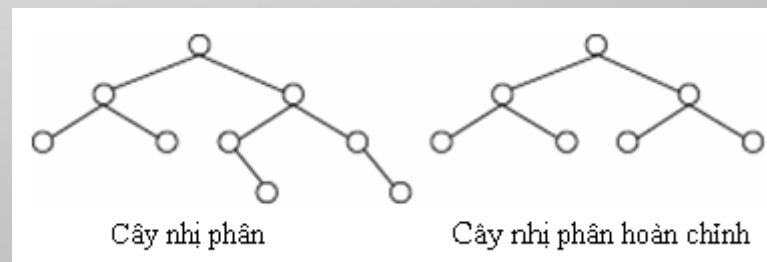


1.3.4 Cây

Cây : Là một cấu trúc dữ liệu biểu diễn cấu trúc phân lớp giữa các phần tử. Nó được sử dụng cho biểu đồ tổ chức của một công ty, cấu hình hệ thống.



một cây mà mỗi nút có không quá 2 con được gọi là một **"cây nhị phân"**. Nếu một cây nhị phân mà tất cả các lá có cùng độ sâu hoặc nếu 2 lá bất kỳ có độ sâu chênh lệch nhỏ hơn hoặc bằng một và các lá hướng sang bên trái, được gọi là **"cây nhị phân hoàn chỉnh"**

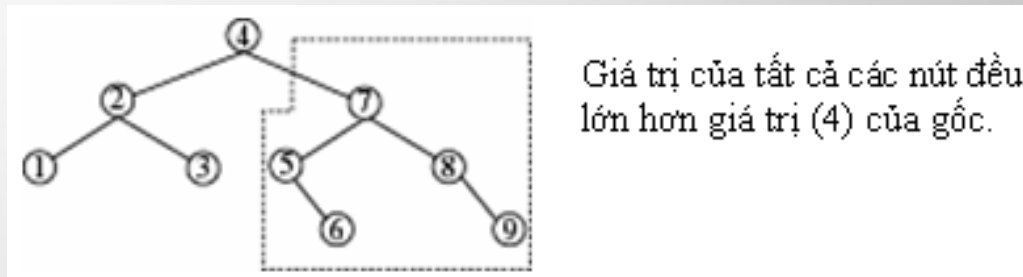


1.3.4. Cây

❑ Cây nhị phân tìm kiếm

Là một cây nhị phân mà giá trị của một phần tử được gán cho mỗi nút thỏa mã các ràng buộc:

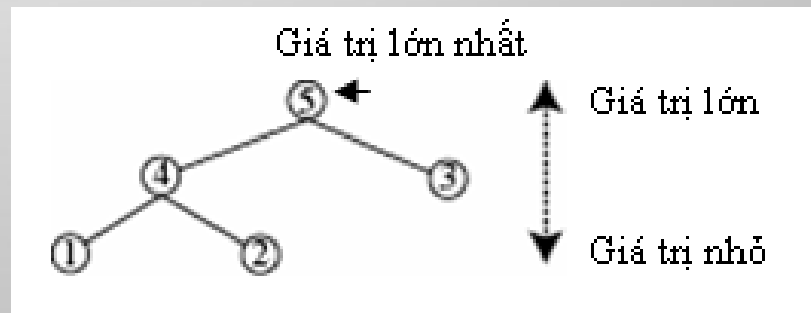
Giá trị con trái < giá trị phần tử cha < giá trị con phải



❑ Đồng

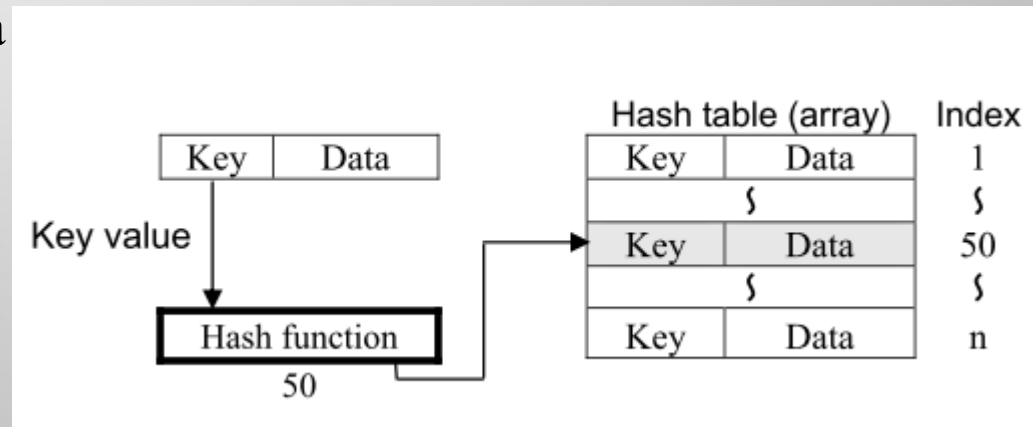
Cây nhị phân được gọi là **đồng** nếu giá trị của nút được gán từ mức gốc và từ trái sang phải trên cùng mức với điều kiện sau:

Giá trị của phần tử cha > giá trị của phần tử con(hoặc ngược lại)



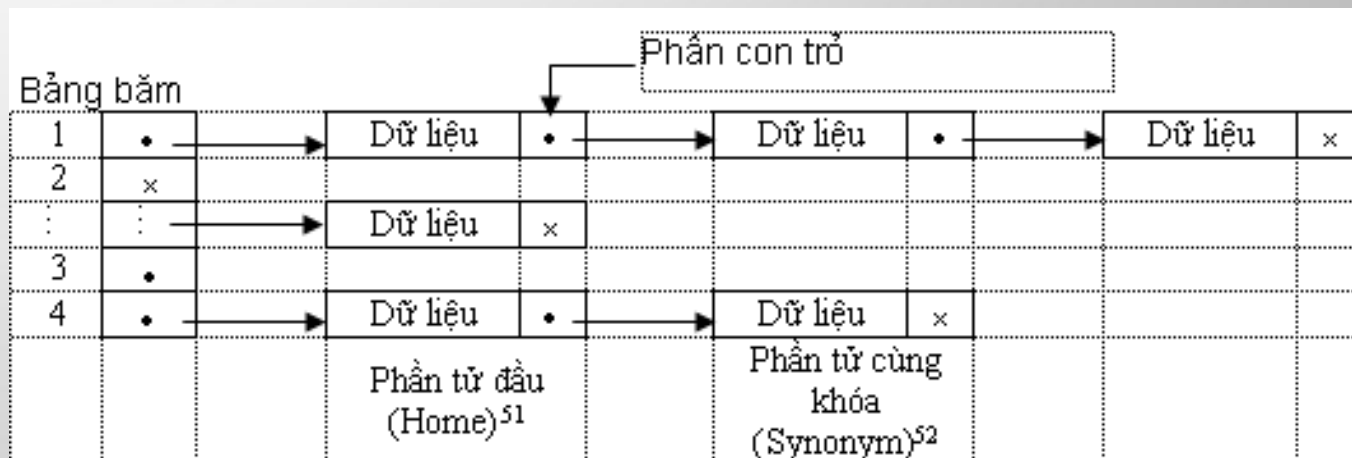
1.3.5 Băm

- ❑ **Băm** là khái niệm sử dụng trực tiếp giá trị khóa như vị trí lưu trữ dữ liệu.
 - Ít khi giá trị khóa được sử dụng trực tiếp như chỉ số.
 - Chuyển giá trị khóa sang chỉ số, **một hàm băm** được sử dụng để tính giá trị băm, được sử dụng như chỉ số.
 - Mảng lưu trữ các phần tử sử dụng phương pháp này gọi là **bảng băm**
 - Khóa liên quan tới nhiều giá trị, chỉ số giống nhau có thể sinh ra từ giá trị khóa khác nhau bằng cách tính chỉ số (giá trị băm) từ 1 đến n sử dụng hàm băm. Khi giá trị băm giống nhau được sinh ra theo cách này được gọi là



1.3.5 Băm

- ❑ Phương pháp dây chuyền (phương pháp băm mở):
Dùng một danh sách để lưu trữ phần tử có giá trị băm giống nhau khi xảy ra xung đột.



QUIZ

- Q1: Ta gọi cấu trúc dữ liệu kiểu “Last-In First-Out (Vào sau ra trước)” là gì?
- Q2: Ta gọi cấu trúc dữ liệu kiểu “First-In First-Out (Vào trước ra trước)” là gì?
- Q4: Định nghĩa “cây nhị phân” và “cây nhị phân đầy đủ”.
- Q4: Ta gọi cây nhị phân thỏa mãn quan hệ sau là gì: “giá trị của con trái < giá trị của phần tử cha < giá trị con phải”?

1.4 Giải thuật

- ❑ Giải thuật tìm kiếm
- ❑ Các giải thuật sắp xếp
- ❑ Các giải thuật tìm kiếm sâu
- ❑ Các giải thuật trên đồ thị
- ❑ QUIZ

1.4.1 Giải thuật tìm kiếm

Một tập hợp các thủ tục để giải quyết một vấn đề gọi là một giải thuật. Hình vẽ biểu thị tập các thủ tục để đạt tới kết quả thích hợp gọi là lưu đồ.

Giải thuật tìm kiếm

- **Tìm kiếm** nghĩa là tìm một phần tử trong một bảng (mảng một chiều)
- Phương pháp tìm kiếm: tìm kiếm tuyến tính (tuần tự) và tìm kiếm nhị phân

Tìm kiếm tuần tự	Tìm kiếm nhị phân
không quan tâm tới các phần tử được sắp xếp như thế nào	yêu cầu các phần tử đã được sắp xếp theo thứ tự tăng dần hoặc giảm dần
Độ phức tạp của phương pháp tìm kiếm tuyến tính = $N / 2$. Số phép so sánh tối đa của phương pháp tìm kiếm tuyến tính = N .	Độ phức tạp của phương pháp tìm kiếm nhị phân = $\lceil \log_2 N \rceil$. Số phép so sánh tối đa của phương pháp tìm kiếm nhị phân = số phép so sánh trung bình + 1.

1.4.2 Giải thuật sắp xếp

- ❑ **Sắp xếp** nghĩa là bố trí lại các phần tử và/hoặc các bản ghi của một mảng theo thứ tự của khóa.
- ❑ Các phương pháp:
 - + sắp xếp nổi bọt.
 - + sắp xếp chọn.
 - + sắp xếp chèn.
 - + sắp xếp nhanh.
 - + sắp xếp trộn.
 - + sắp xếp đóng

1.4.2 Giải thuật sắp xếp

Sắp xếp nổi bọt: mỗi cặp phần tử liên nhau được so sánh tuần tự và đổi chỗ nếu cần thiết.

Trước khi sắp xếp	5	4	3	2	1:	
Lần 1	5 ⇌ 4	3	2	1:	Đổi chỗ 5 và 4	
	4	5 ⇌ 3	2	1:	Đổi chỗ 5 và 3	
	4	3	5 ⇌ 2	1:	Đổi chỗ 5 và 2	
	4	3	2	5 ⇌ 1:	Đổi chỗ 5 và 1	
	4	3	2	1	5:	Kết thúc lần 1 (giá trị lớn nhất nằm cuối cùng bên phải)
Lần 2	4 ⇌ 3	2	1	5:	Đổi chỗ 4 và 3 (giá trị bên phải của " " đã được sắp xếp)	
	3	4 ⇌ 2	1	5:	Đổi chỗ 4 và 2	
	3	2	4 ⇌ 1	5:	Đổi chỗ 4 và 1	
	3	2	1	4	5:	Kết thúc lần 2 (giá trị lớn thứ hai nằm ở vị trí thứ hai từ bên phải)
Lần 3	3 ⇌ 2	1	4	5:	Đổi chỗ 3 và 2 (giá trị bên phải của " " đã được sắp xếp)	
	2	3	1	4	5:	Đổi chỗ 1 và 3
	2	1 ⇌ 3	4	5:	Kết thúc lần 3 (giá trị lớn thứ ba nằm ở vị trí thứ ba từ bên phải)	
Lần 4	2 ⇌ 1	3	4	5:	Đổi chỗ 2 và 1 (giá trị bên phải của " " đã được sắp xếp)	
	1	2	3	4	5:	Kết thúc lần 4, sắp xếp xong

1.4.2 Giải thuật sắp xếp

Sắp xếp chọn:

- Tìm giá trị lớn nhất (hoặc nhỏ nhất) từ mảng và thay đổi nó với phần tử nằm ở cuối mảng.
- Tìm phần giá trị lớn nhất (hoặc nhỏ nhất) từ mảng trừ phần tử cuối cùng và đổi vị trí của nó với phần tử thứ hai từ cuối lên của mảng.
- Lặp lại thủ tục này, sắp xếp chọn kết thúc khi nó so sánh phần tử đầu tiên và phần tử thứ hai của mảng.

Lần 1	5	4	3	2	1:	Vì "5" là giá trị lớn nhất, nó đổi chỗ với phần tử cuối cùng "1".
Lần 2	1	4	3	2	5:	Vì 4 là giá trị lớn nhất, nó đổi chỗ cho phần tử cuối cùng trong lần chạy thứ 2.
Lần 3	1	2	3	4	5:	Sắp xếp xong

1.4.2 Giải thuật sắp xếp

Sắp xếp chèn:

- Bắt đầu với 1 mảng đã được sắp xếp, so sánh phần tử cần chèn với các phần tử trong mảng từ đầu tới cuối, và chèn phần tử vào vị trí thích hợp
- Ví dụ dưới, các phần tử bên trái “|” đã được sắp xếp, ở đây chỉ có duy nhất 1 phần tử trong lần chạy đầu tiên. Nó coi như đã được sắp xếp.

Lần 1	5 4 3 2	1:	Vì 4 là giá trị nhỏ nhất, nó được chèn vào vị trí thích hợp (trước 5)
Lần 2	4 5 3 2	1:	Vì 3 là giá trị nhỏ nhất, nó được chèn vào vị trí thích hợp (trước 4)
Lần 3	3 4 5 2	1:	Vì 2 là giá trị nhỏ nhất, nó được chèn vào vị trí thích hợp (trước 3)
Lần 4	2 3 4 5	1:	Vì 1 là giá trị nhỏ nhất, nó được chèn vào vị trí thích hợp (trước 2)
	1 2 3 4 5	5:	Sắp xếp xong

1.4.2 Giải thuật sắp xếp

Sắp xếp nhanh

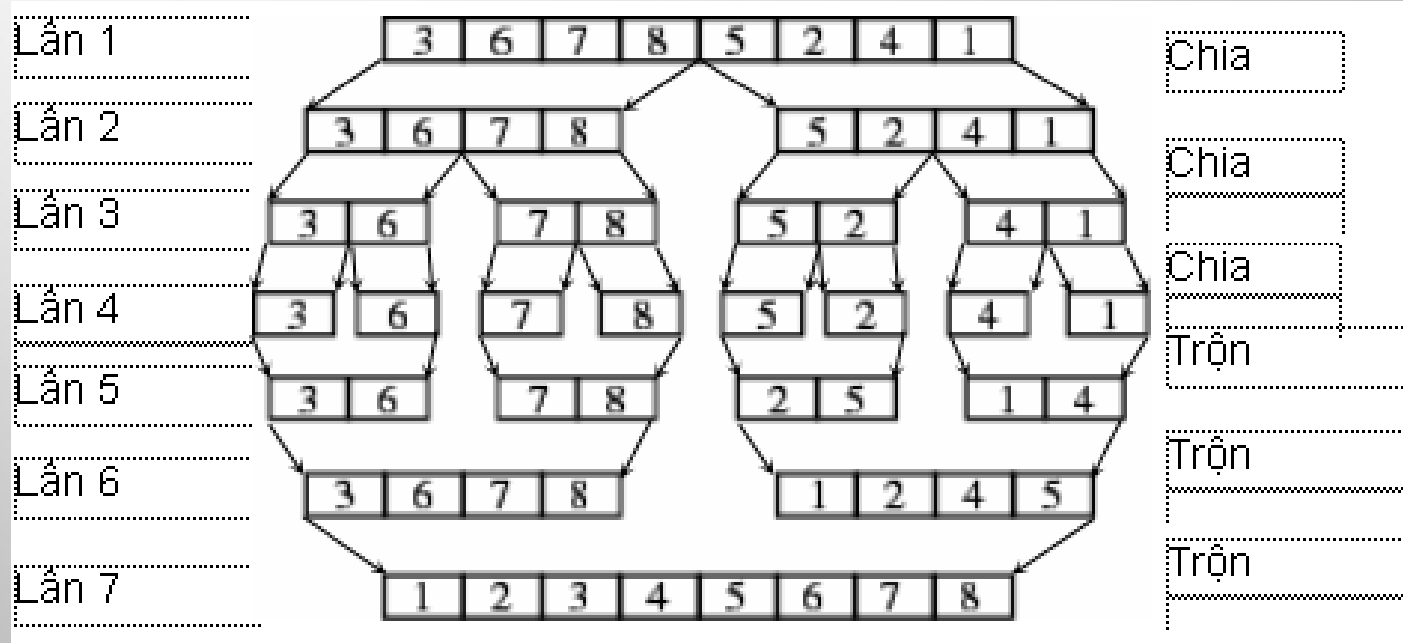
- Lựa chọn 1 giá trị ngẫu nhiên từ mảng và sử dụng nó như 1 chốt .
- Các phần tử được chia thành 2 nhóm, nhóm đầu tiên chứa tất cả các phần tử nhỏ hơn chốt và nhóm thứ hai chứa tất cả các phần tử lớn hơn chốt (giá trị bằng có thể đặt ở 1 trong 2 nhóm).
- Thủ tục tương tự được lặp đi lặp lại cho mỗi nhóm. Nó tiếp tục cho đến khi chỉ còn duy nhất 1 phần tử trong mỗi nhóm.

Lần 1	2	5	6	4	1	<u>3</u>	Chia thành 2 khối
Lần 2	2	1	<u>3</u>	5	6	<u>4</u>	Chia mỗi khối làm 2
Lần 3	1	<u>2</u>	<u>3</u>	4	5	<u>6</u>	Chia mỗi khối làm 2 (trừ các nhóm chỉ có 1 phần tử)
Lần 4	1	2	3	4	5	6	Sắp xếp xong

1.4.2 Giải thuật sắp xếp

Sắp xếp trộn

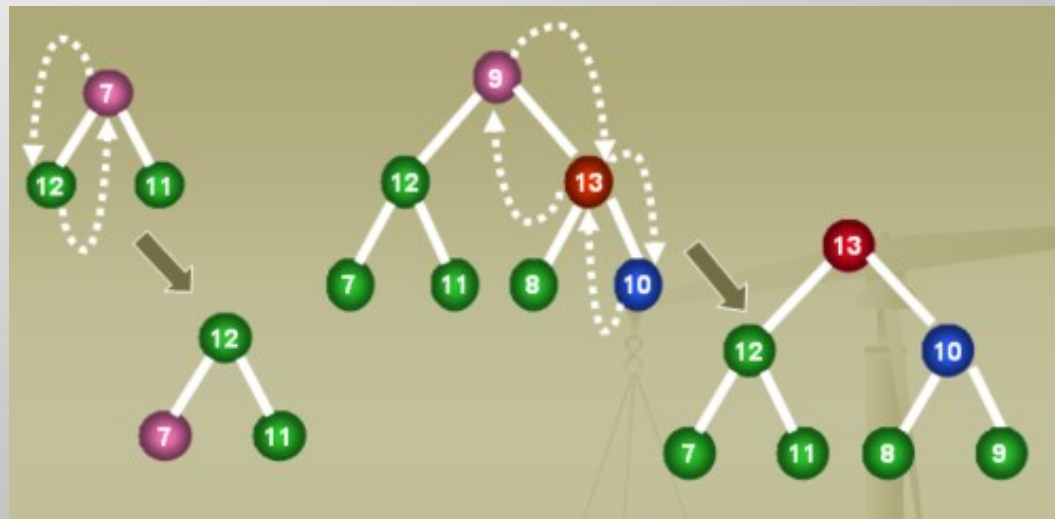
- phép chia nhóm được lặp lại đến khi mỗi nhóm chỉ có duy nhất 1 phần tử.
- Tiếp đó các phần tử được trộn vào nhau tuần tự.



1.4.2 Các giải thuật sắp xếp

Sắp xếp vun đống

- Đống là 1 cây nhị phân trong đó mọi cây con có tính chất: phần tử cha có giá trị lớn hơn các con của nó.
- Nếu phần tử gốc được chọn, ta có thể thu được giá trị lớn nhất trong khi các phần tử còn lại có thể tái cấu trúc thành đống mới.
- Bằng cách lặp lại thao tác lấy gốc và tái cấu trúc đống, sắp xếp có thể hoàn tất.



1.4.3 Giải thuật tìm kiếm sâu

Tìm kiếm sâu

- Là quá trình tìm một chuỗi kí tự đặc biệt trong một văn bản (xâu kí tự).
- Các phương pháp: phương pháp duyệt toàn bộ, phương pháp Boyer-Moore...

4.3.1. Phương pháp duyệt toàn bộ:

Phương pháp trong đó xâu cần tìm được bằng cách so sánh từng kí tự từ đầu dãy.

4.3.2. Phương pháp Boyer-Moore:

- Phương pháp lấy nội dung của mẫu trong văn bản bỏ qua những phần vô giá trị.
- Nếu mẫu và 1 xâu của văn bản không trùng khớp, số kí tự có thể nhảy qua dựa trên kí tự ngoài cùng bên phải của khoảng tìm kiếm của văn bản được so sánh.

1.4.3 Giải thuật tìm kiếm sâu

Ta giải thích chi tiết dùng ví dụ như trong tìm kiếm vết cạn

- (1) Nếu kí tự ngoài cùng bên phải của phần văn bản đang được so sánh với chuỗi là “X”, vị trí tiếp theo có khả năng mà mẫu có thể trùng là sau đó 2 kí tự, nên 2 kí tự tiếp theo được nhảy qua
- (2) Nếu kí tự ngoài cùng bên phải của phần văn bản đang được so sánh với chuỗi là “Y”, vị trí tiếp theo có khả năng mà mẫu có thể trùng là sau đó 1 kí tự, nên kí tự tiếp theo được nhảy qua
- (3) Nếu kí tự ngoài cùng bên phải của phần văn bản đang được so sánh với chuỗi là “Z”, vị trí tiếp theo có khả năng mà mẫu có thể trùng là sau đó 3 kí tự, nên 3 kí tự tiếp theo được nhảy qua

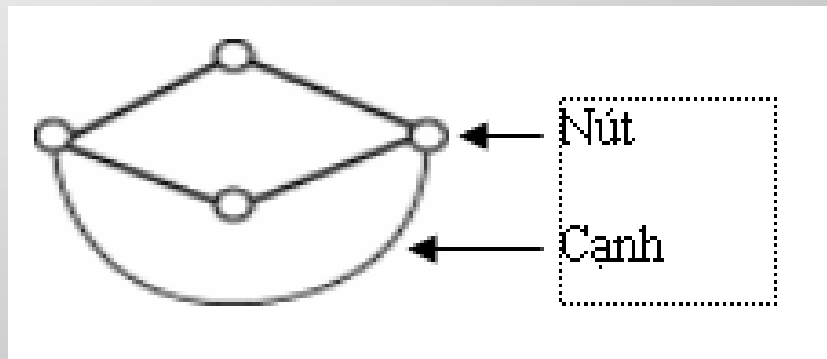
Văn bản	***X***	Văn bản	***Y***	Văn bản	***Z***
Trước khi di chuyển	XYZ	Trước khi di chuyển	XYZ	Trước khi di chuyển	XYZ
Sau khi di chuyển	→XYZ	Sau khi di chuyển	→XYZ	Sau khi di chuyển	→→XYZ

- (4) Nếu kí tự ngoài cùng bên phải của phần văn bản không phải là X, Y hoặc Z thì tình huống giống hệt với (3), nên 3 kí tự tiếp theo được nhảy qua⁶²

1.4.4 Tìm kiếm trên đồ thị

Là giải thuật mà phép tìm kiếm được thực hiện trên một cây

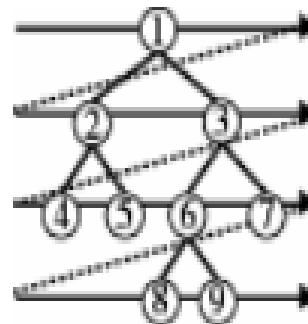
- Bao gồm tìm kiếm theo chiều rộng hoặc chiều sâu.
- Đồ thị gồm các nút và cạnh. Nút là một đỉnh của đồ thị và cạnh là đoạn nối giữa 2 đỉnh.
- Một cây có thể coi là một đồ thị trong đó không phải tất cả các nút được nối với tất cả các nút khác.



1.4.4 Tìm kiếm trên đồ thị

Thứ tự ưu tiên chiều rộng

Phép tìm kiếm bắt đầu ở gốc và duyệt ngang qua các nút ở mức thấp từ trái sang phải. Trong ví dụ dưới đây, số ở các nút xác định thứ tự mà nút đó được duyệt qua.



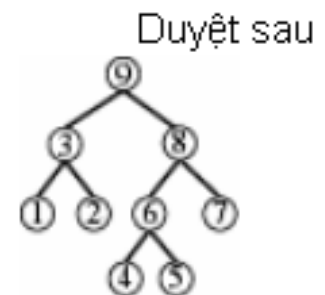
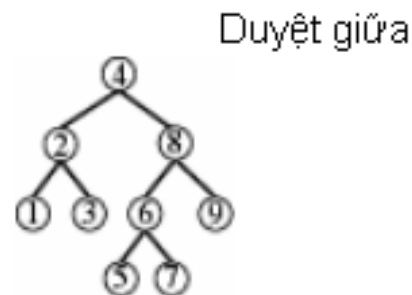
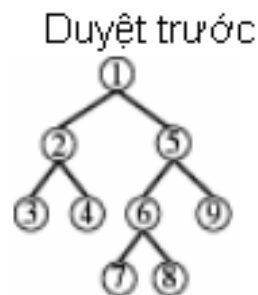
1.4.4 Tìm kiếm trên đồ thị

Thứ tự ưu tiên chiều sâu

Trong tìm kiếm ưu tiên chiều sâu⁶⁴, chúng ta bắt đầu ở gốc, duyệt từ cây con trái rồi từ lá. Dựa trên thời điểm mà các nút được duyệt, ta có thể chia lớp như bảng dưới

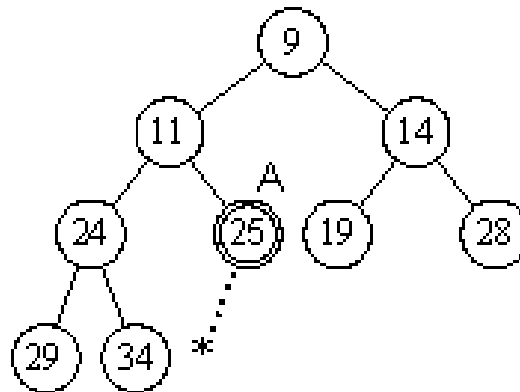
Phương pháp duyệt	Thứ tự duyệt các nút
Duyệt trước	Thứ tự: cha, con trái, con phải
Duyệt giữa	Thứ tự: con trái, cha, con phải
Duyệt sau	Thứ tự: con trái, con phải, cha

Trong hình dưới, số trên các nút xác định thứ tự mà nút được duyệt



QUIZ

- Q1:** Trong tìm kiếm nhị phân, khi số lượng dữ liệu đã sắp xếp tăng gấp 4 lần thì số lượng phép so sánh tối đa tăng bao nhiêu?
- Q2:** Cho đồng dưới đây, giá trị của nút cha nhỏ hơn giá trị của các nút con. Khi chèn 1 nút vào đồng này, 1 phần tử được thêm vào ở cuối cùng. Nếu phần tử đó nhỏ hơn nút cha, cha và con phải chuyển chỗ cho nhau. Nếu phần tử 7 được thêm vào đồng ở vị trí được đánh dấu (*), phần tử nào sẽ nằm ở vị trí A.



a) 7

b) 11

c) 24

d) 25

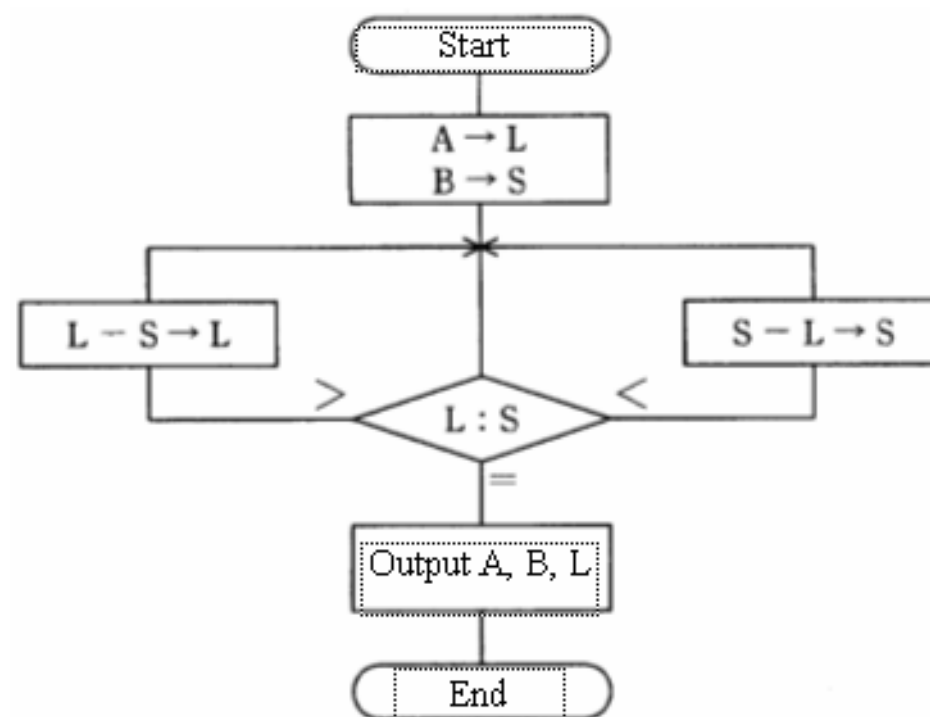
Câu hỏi ôn tập

Q1. Bảng quyết định dưới đây để tạo các báo cáo từ tệp nhân viên. Điều gì dưới đây có thể kết luận từ bảng quyết định này

Dưới 30 tuổi	Y	Y	N	N
Nam	Y	N	Y	N
Đã kết hôn	N	Y	Y	N
Xuất ra báo cáo 1	–	X	–	–
Xuất ra báo cáo 2	–	–	–	X
Xuất ra báo cáo 3	X	–	–	–
Xuất ra báo cáo 4	–	–	X	–

- a) Báo cáo 1 chứa nội dung của Báo cáo 4 trừ dữ liệu của nam từ 30 tuổi trở lên
- b) Báo cáo 2 chứa tất cả nam chưa lấy vợ
- c) Nam trong báo cáo 3 nằm trong báo cáo 2
- d) Những người trong báo cáo 4 không nằm trong bất kì báo cáo nào khác

- Q2. Lưu đồ dưới minh họa giải thuật Euclide tìm ước chung lớn nhất của A và B bằng các lặp lại phép trừ. Khi A là 876 và B là 204, có bao nhiêu phép so sánh được thực hiện trong quá trình này



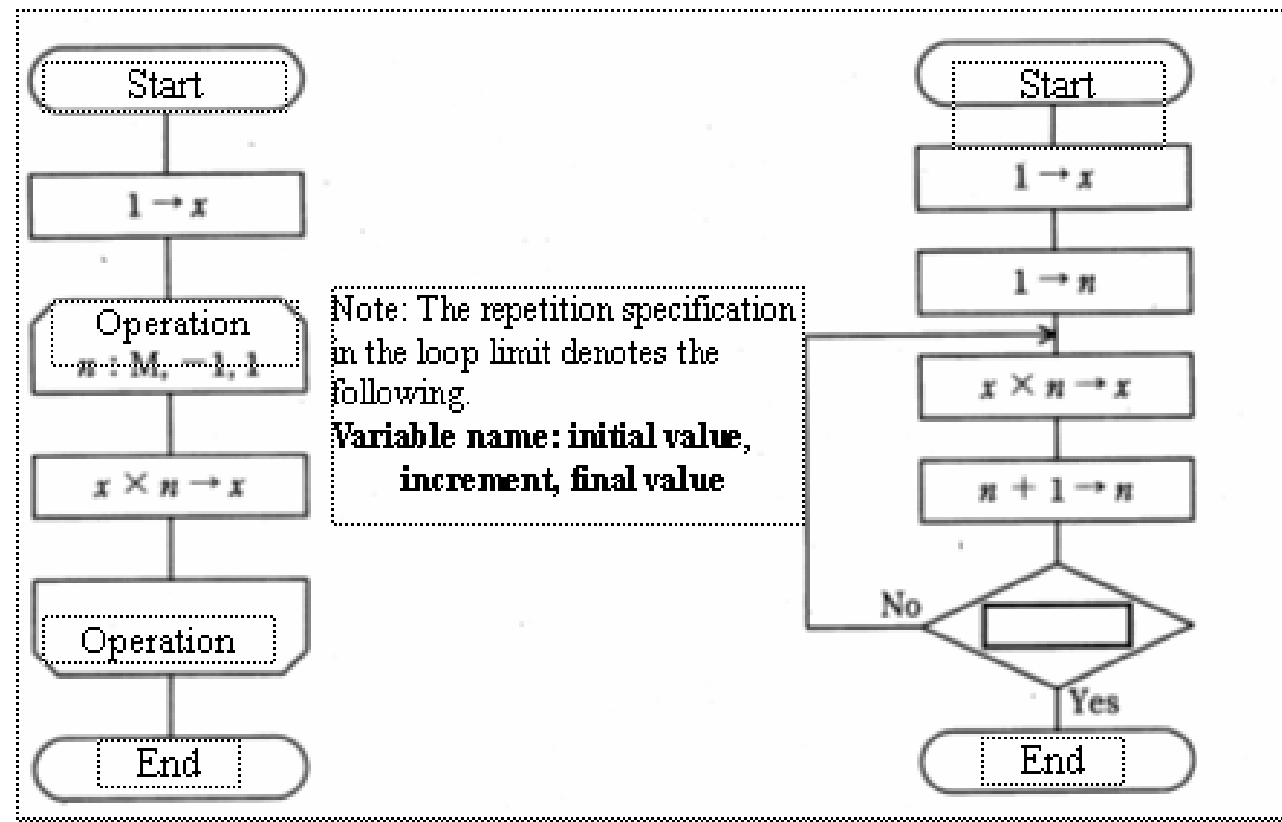
a) 4

b) 9

c) 10

d) 11

Các giải thuật mô tả bằng 2 lưu đồ dưới đây được thực hiện trên 1 số nguyên dương M , điều kiện nào cần chèn thêm vào ô bên dưới sao cho nhận được cùng kết quả x?



a) $n > M$

b) $n > M + 1$

c) $n > M - 1$

d) $n < M$