

# 자료구조론

## 1장 자료구조 개요

## □ 이 장에서 다룰 내용

- ❖ 자료구조 개요
- ❖ 자료구조의 분류
- ❖ 자료의 표현

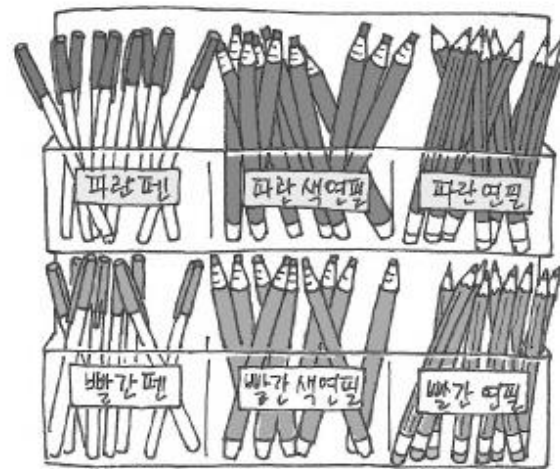
## □ 자료구조 개요

### ❖ 자료구조란?

- Data Structure
- 자료를 효율적으로 사용할 수 있도록 자료를 특성에 따라서 분류하여 저장하는 방법과 자료에 대한 연산을 정의



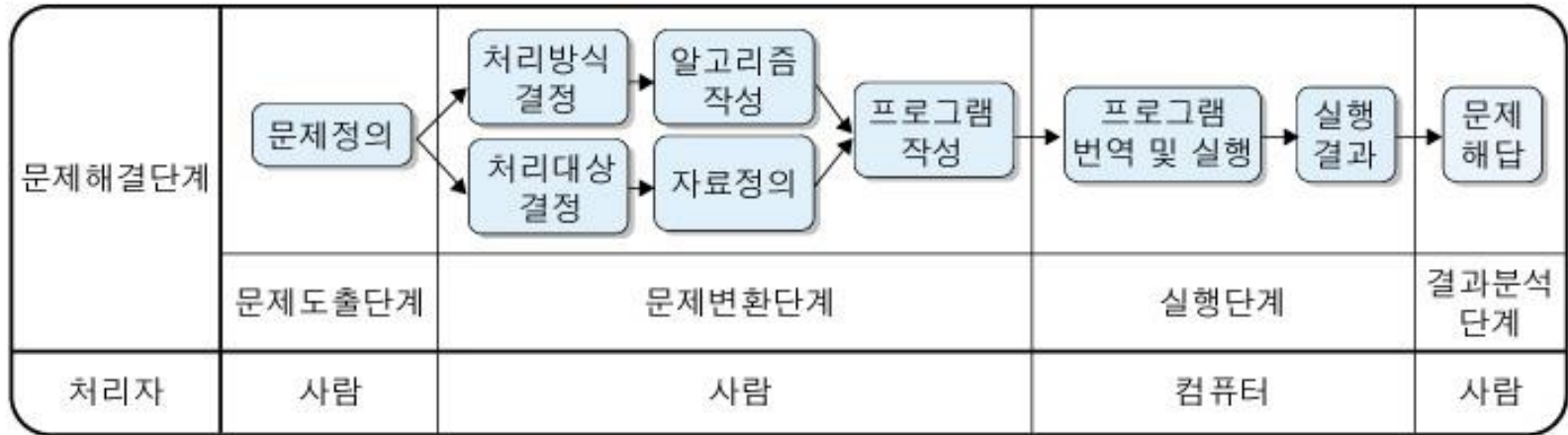
[나쁜 자료구조]



[좋은 자료구조]

## □ 자료구조 개요

### ❖ 컴퓨터에 의한 문제 해결 과정



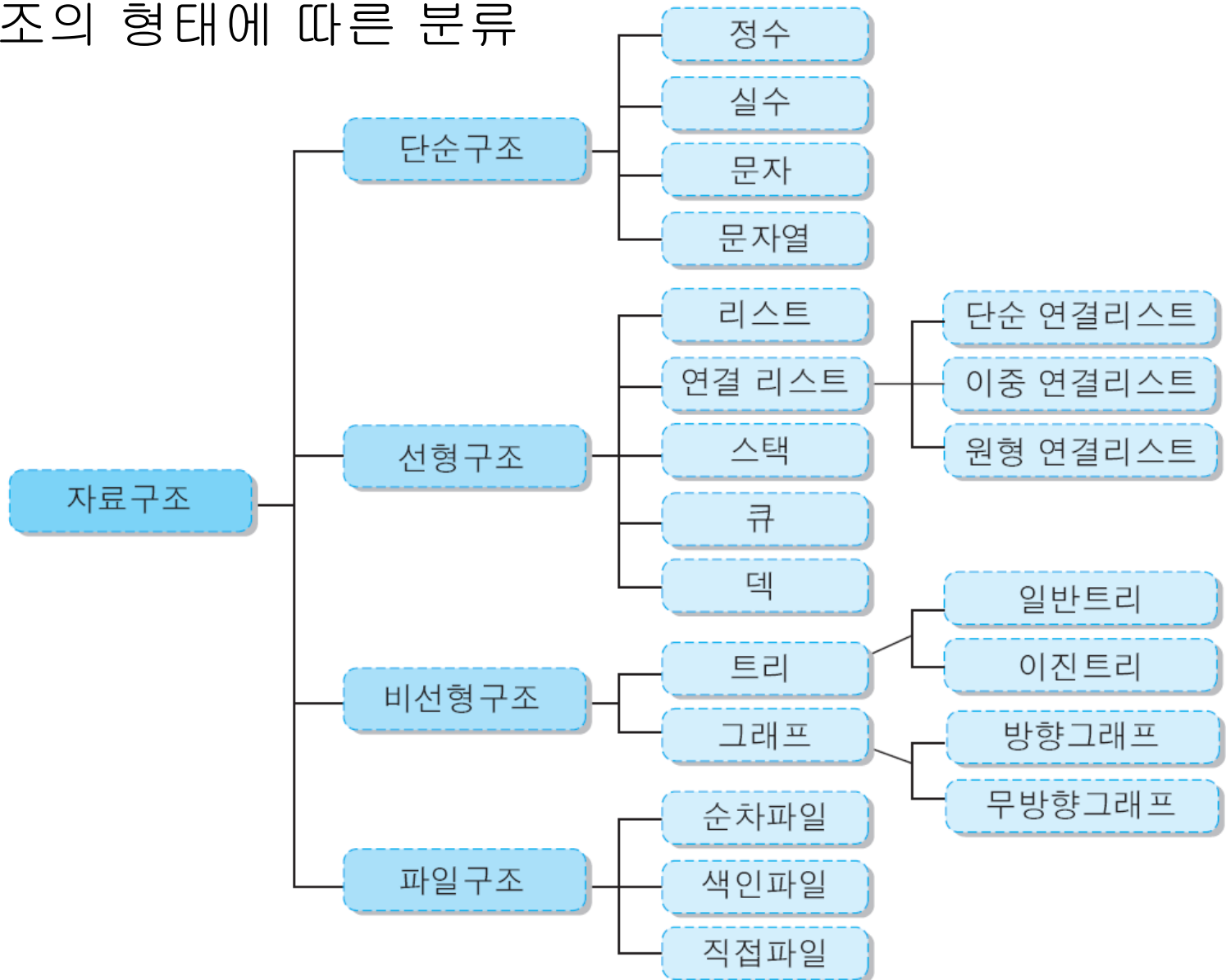
- 컴퓨터가 효율적으로 문제를 처리하기 위해서는 문제를 정의하고 분석하여 그에 대한 최적의 프로그램을 작성해야 한다.
- 프로그램은 자료를 표현하고, 그 자료를 처리하여 결과를 얻음  

자료구조
알고리즘

➔ 다양한 자료구조에 대한 이해와 활용 능력이 필요함

## □ 자료구조의 분류

### ❖ 자료구조의 형태에 따른 분류



# □ 자료구조의 분류

## ❖ 자료구조의 형태에 따른 분류

- 단순 구조
  - 정수, 실수, 문자, 문자열 등의 기본 자료형
- 선형 구조
  - 자료들 간의 앞뒤 관계가 1:1의 선형 관계
  - 리스트, 연결리스트, 스택(stack), 큐(queue), 덱(deque) 등
- 비선형 구조
  - 자료들 간의 앞뒤 관계가 1:多, 또는 多:多의 관계
  - 계층(hierarchical)구조, 망(network)구조
  - 트리, 그래프 등
- 파일 구조
  - 레코드의 집합인 파일에 대한 구조
  - 순차파일, 색인파일, 직접파일 등

## □ 자료구조의 분류

### ❖ 자료구조의 선택

■ 예: 정수 리스트에서 25를 찾으시오.

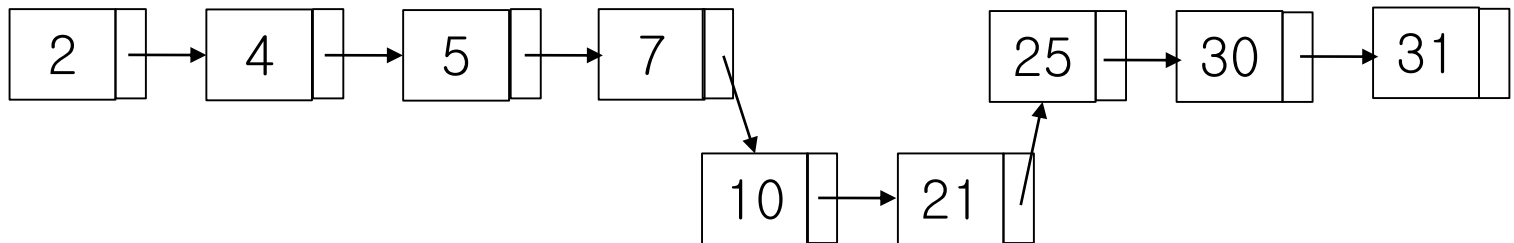
- 순차 리스트 1

10	30	21	4	31	5	2	25	7
----	----	----	---	----	---	---	----	---

- 순차 리스트 2

2	4	5	7	10	21	25	30	31
---	---	---	---	----	----	----	----	----

- 연결 리스트



## □ 자료구조의 분류

### ❖ 자료구조의 선택

- 자료구조 선택의 중요성
  - 자료구조는 처리하고자 하는 자료의 표현과 저장 방법을 결정
  - 자료구조와 알고리즘은 밀접한 관계
  - 따라서 어떤 자료구조를 선택하느냐에 따라 처리 능률이 크게 달라짐
- 자료구조 선택의 기준
  - 자료의 양
  - 자료의 활용 빈도
  - 자료의 갱신 정도
  - 자료 처리를 위해 사용 가능한 기억장치 용량
  - 자료 처리 시간의 제한
  - 자료 처리를 위한 프로그래밍의 용이성



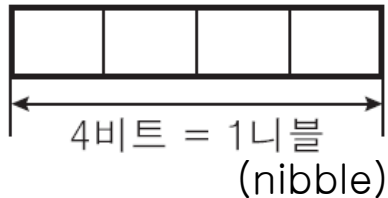
## □ 자료의 표현

### ❖ 디지털 시스템에서 자료의 표현

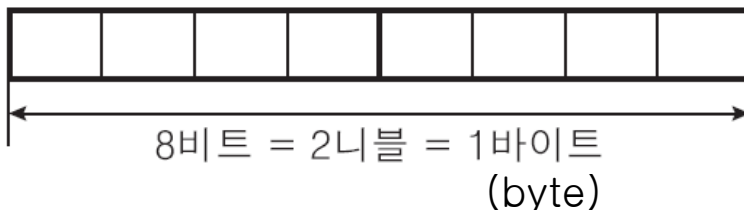
- 숫자, 문자, 그림, 소리, 기호 등 모든 형식의 자료를 2진수 코드로 표현하여 저장 및 처리
- 2진수 코드
  - 1과 0, 또는 ON과 OFF, 또는 참(true)과 거짓(false)의 조합
- 2진수 코드의 단위



➔ 자료 표현하는 최소 단위



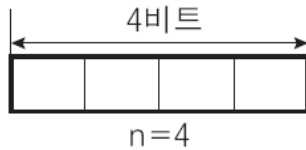
➔ 16진수 표현



➔ 자료 처리/저장/전송의 기본 단위

## □ 자료의 표현

- $n$ 개의 비트로  $2^n$ 개의 상태수 표현
  - 예)  $n=4$  인 경우

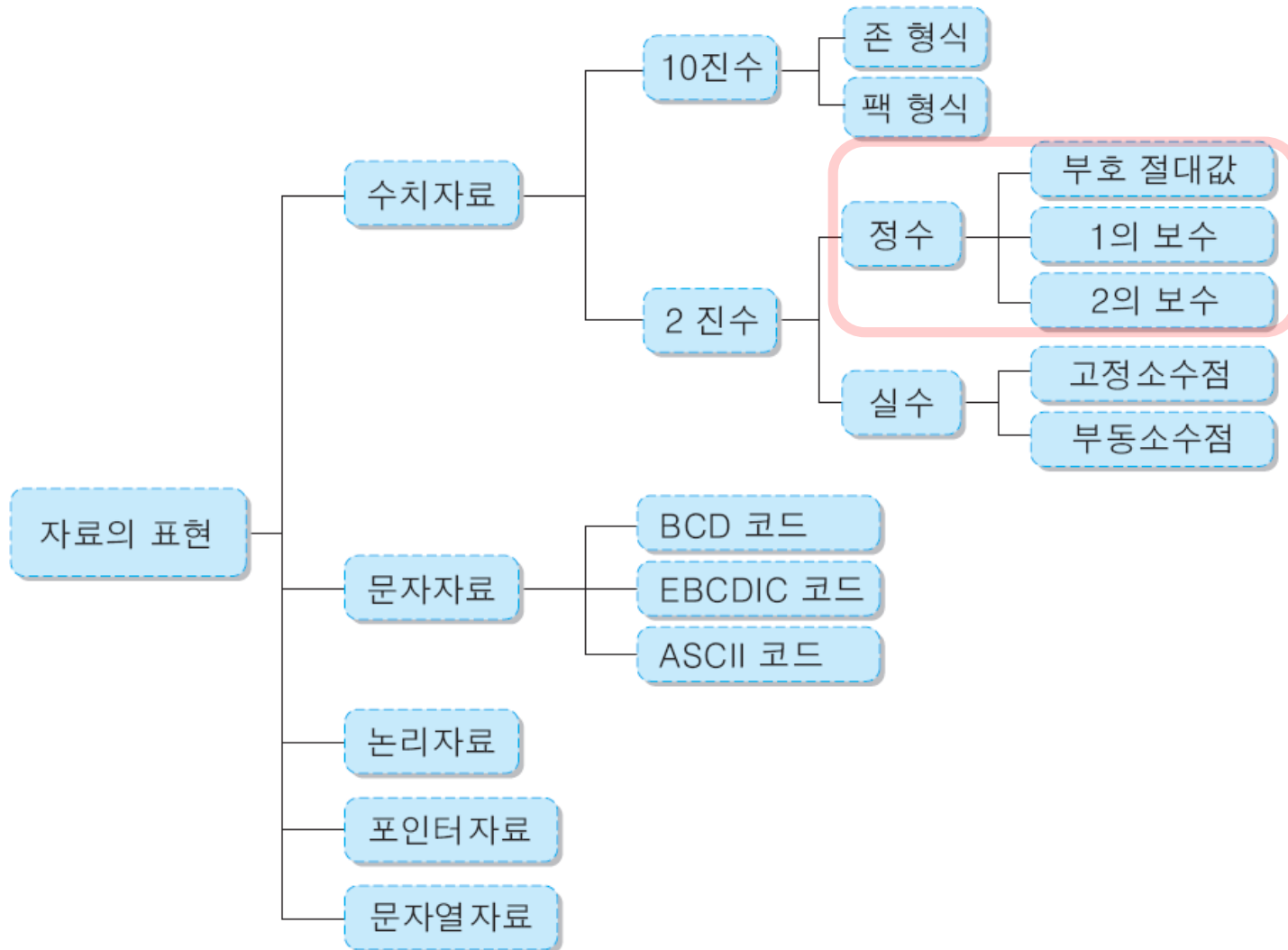


0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

16개 ( $2^n = 2^4 = 16$ )의 상태값 표현

## □ 자료의 표현

### ❖ 컴퓨터 내부에서 자료를 표현하는 방법



## □ 자료의 표현 - 2진수 정수

0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

## □ 자료의 표현 - 2진수 정수

### ❖ 2진수 정수 표현 : n비트의 부호 절대값 형식

- 최상위 1비트 → 부호 표시
  - 양수(+) : 0
  - 음수(-) : 1
- 나머지 n-1 비트 → 절대값의 이진수 표시
- 예) 1 바이트 정수인 경우

+ 21

1비트	← 7 비트 →						
0	0	0	1	0	1	0	1
부호	절대값 = 21						

- 21

1비트	← 7 비트 →						
1	0	0	1	0	1	0	1
부호	절대값 = 21						

## □ 자료의 표현 - 2진수 정수

### ❖ 2진수 정수 표현 : 1의 보수 형식

- 음수 표현에 부호 비트를 사용하는 대신 1의 보수(1's complement)를 사용하는 방법
- 예) 21의 1의 보수 구하기 (1 바이트 정수인 경우)

$$\begin{array}{r} 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \\ -\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 1 \\ \hline 1\ 1\ 1\ 0\ 1\ 0\ 1\ 0 \end{array}$$

☞ 21

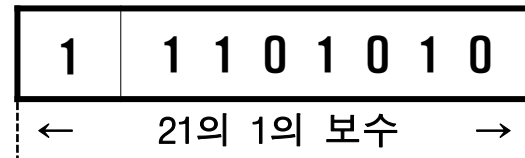
☞ 21의 1의 보수 = -21

+ 21



☞ 부호절대값형식의  
양수 표현과 같음!

- 21



## □ 자료의 표현 - 2진수 정수

## 2진수 정수 표현 : 2의 보수 형식

- 음수 표현에 2의 보수(2's complement)를 사용
- 예) 21의 1의 보수 구하기 (1 바이트 정수인 경우)

1의 보수에 1을 더하면  
2의 보수

$$\begin{array}{r}
 \phantom{-} 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \\
 - 0\ 0\ 0\ 1\ 0\ 1\ 0\ 1 \\
 \hline
 1\ 1\ 1\ 0\ 1\ 0\ 1\ 0 \\
 + \phantom{0000000} 1 \\
 \hline
 1\ 1\ 1\ 0\ 1\ 0\ 1\ 1
 \end{array}$$

➡ 21

☞ 21의 1의 보수

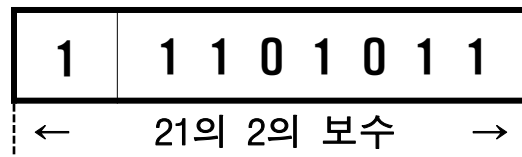
👉 21의 2의 보수 = -21

+ 21



☞ 부호절대값형식의 양수 표현과 같음!

– 21



→ 2진수 정수의 세 가지 표현 방법에서 양수의 표현은 같고, 음수의 표현만 다르다.

## □ 자료의 표현 - 2진수 정수

### ❖ 2진수 정수 표현 : 2의 보수 형식

- 8 비트로 표현 가능한 정수의 범위

$$-2^7 \sim 2^7 - 1 \quad \rightarrow 2^8 \text{ 가지 정보}$$

$$\text{즉, } -128 \sim 127$$

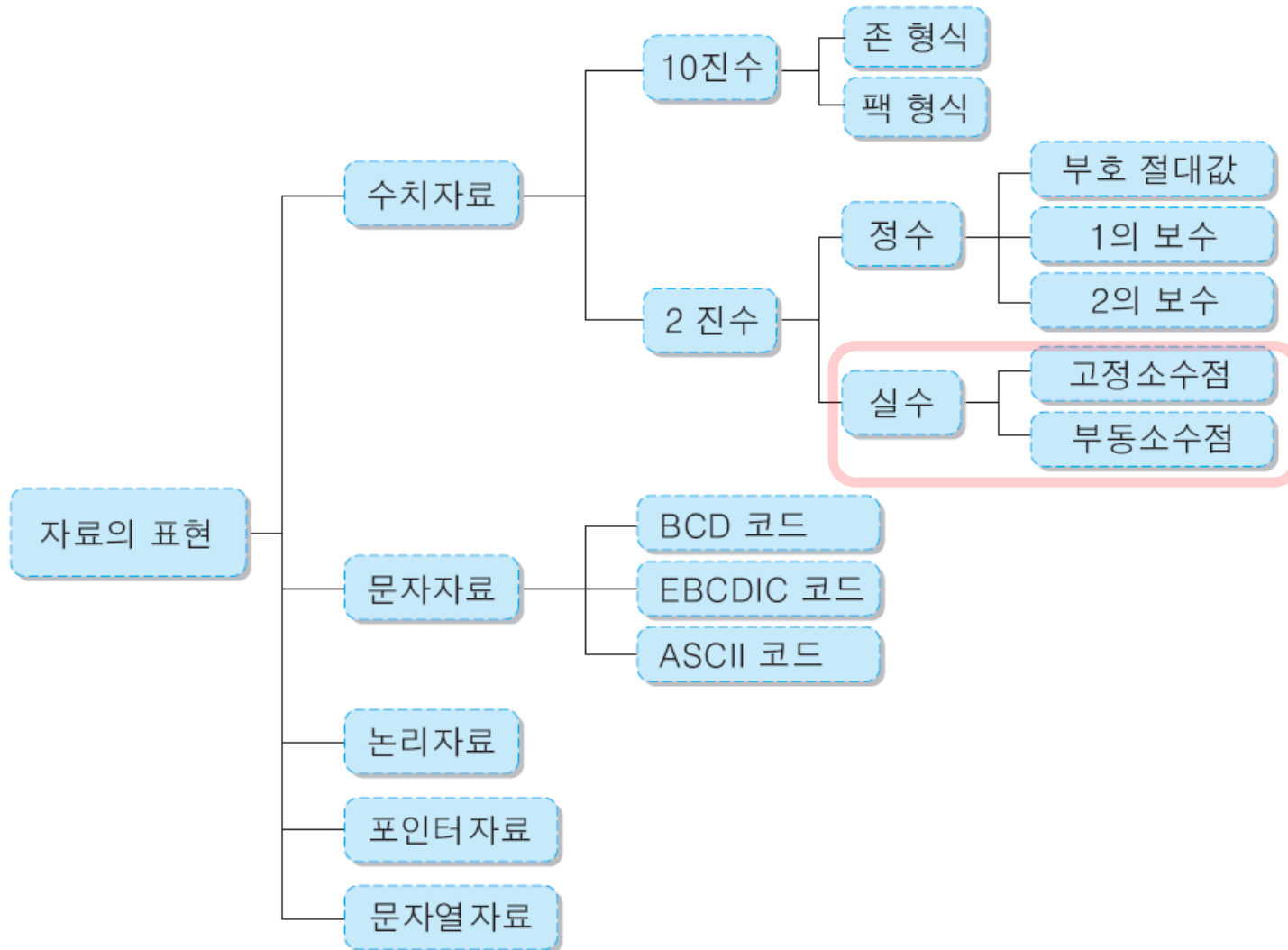
- n 비트로 표현 가능한 정수의 범위

$$-2^{n-1} \sim 2^{n-1} - 1 \quad \rightarrow 2^n \text{ 가지 정보}$$



## □ 자료의 표현

### ❖ 컴퓨터 내부에서 자료를 표현하는 방법



## □ 자료의 표현 - 2진수 실수

### ❖ 2진수 실수 표현 : 고정 소수점(fixed point) 형식

- 소수점이 항상 최상위 비트의 왼쪽 밖에 고정된 것으로 취급
- 고정 소수점 표현의 00011101은 0.00011101의 실수 값을 의미
- 표현 가능한 값의 범위가 작음
  - 예를 들어 위의 고정소수점 표현으로는
    - 0.00000011101 은 00000011 로 표현됨
    - 0.00000000011101 은 00000000 로 표현됨

## □ 자료의 표현 - 2진수 실수

❖ 2진수 실수 표현 : 부동 소수점(floating point) 형식

- 소수점의 위치가 고정되어 있지 않고 변함
- 실수를 **부호**, **지수부**, **가수부**의 세 부분으로 구분하여 표현
  - 부동 소수점 개념 이해를 돕기 위해 10진법 예를 들어 설명한다.

$$-0.0002345 = -0.2345 * 10^{-3}$$

부호                      가수(mantissa, 유효숫자)                      지수(exponent)                      밑수(base, radix)

1	-3	2345
부호	지수부	가수부

$$23450000 = 0.2345 * 10^8$$

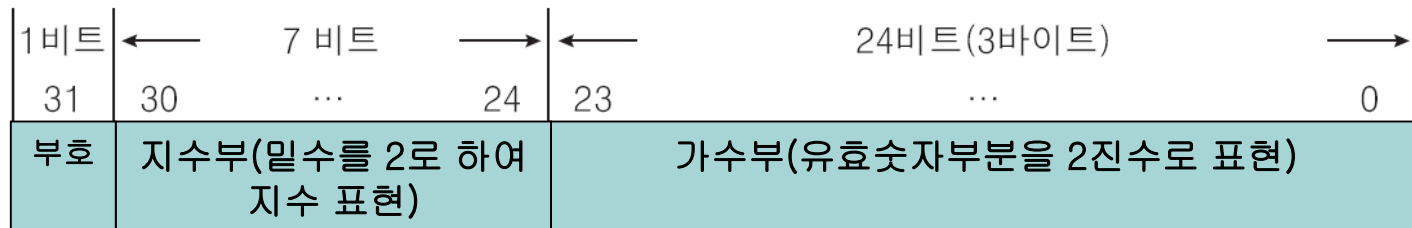
0	8	2345
---	---	------

$$0.00000002345 = 0.2345 * 10^{-7}$$

0	-7	2345
---	----	------

## □ 자료의 표현 - 2진수 실수

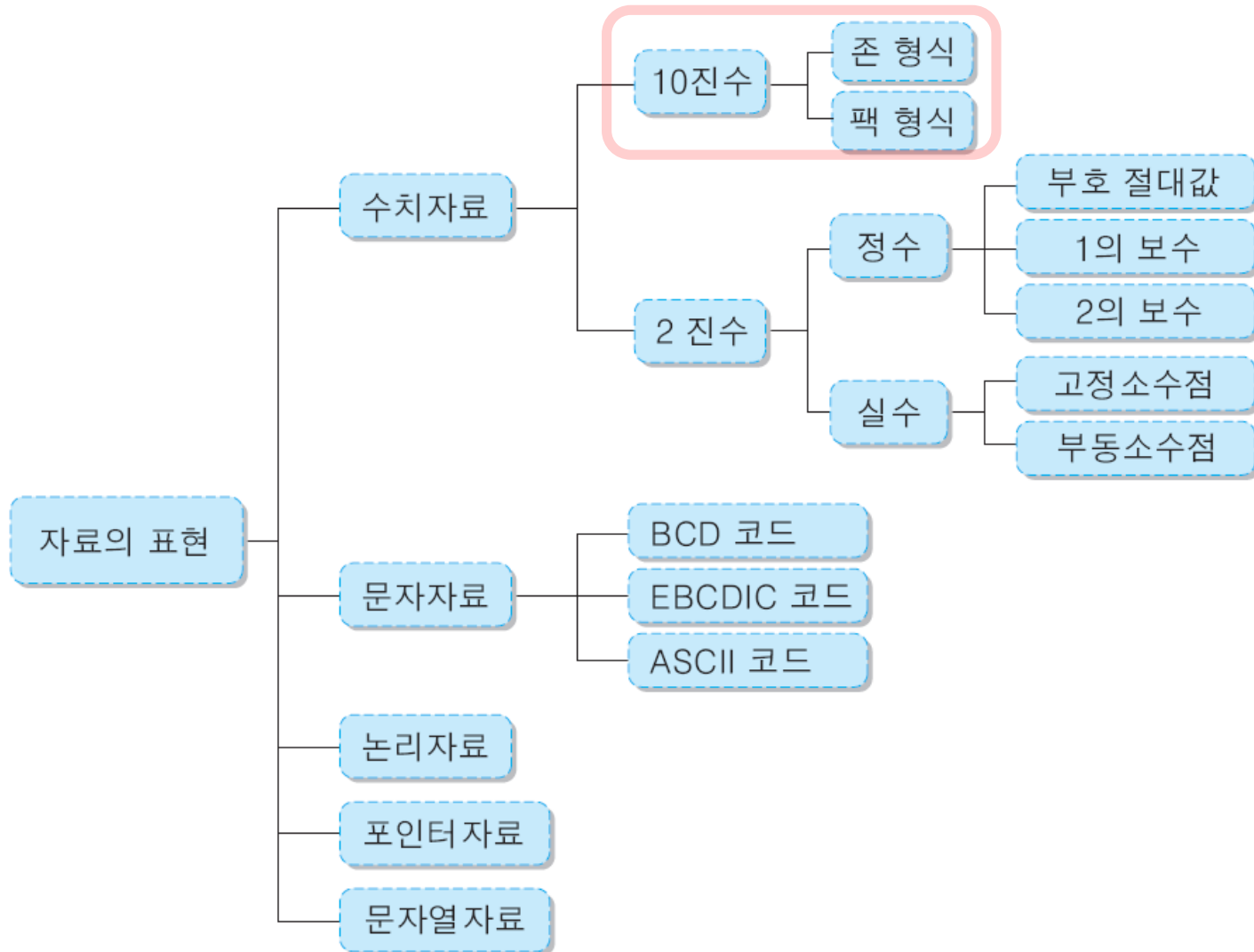
- 부동 소수점 형식 (4바이트 사용하는 경우 예)
  - 실제로는 10진법이 아니라 2진법으로 표현됨



- 부동 소수점 형식은 고정 소수점 형식에 비해서 표현 가능한 값의 범위가 넓다.

## □ 자료의 표현

### ❖ 컴퓨터 내부에서 자료를 표현하는 방법



## □ 자료의 표현 - 10진수

- ❖ 부동 소수점 형식은 근본적으로 부정확한 표현법임
  - 정확한 정밀도(exact precision)를 갖도록 실수값을 표현하려면 2진수 표현 대신 10진수 표현법을 사용해야 함
- ❖ 10진수 자료를 표현하는 방법
  - 존 형식(zoned decimal)
  - 팩 형식(packed decimal)

## □ 자료의 표현 - 10진수

### ❖ 존 형식의 10진수(zoned decimal)

- 10진수 한자리를 표현하기 위해서 1 바이트(8 bits)를 사용
  - 존 영역(상위 니블)
    - 1111로 표시
  - 수치 영역(하위 니블)
    - 표현하고자 하는 10진수 한자리 값에 대한 2진수 값을 표시
- 존 형식의 구조

존 영역				수치 영역			
				8	4	2	1
x	x	x	x	x	x	x	x

## □ 자료의 표현 - 10진수

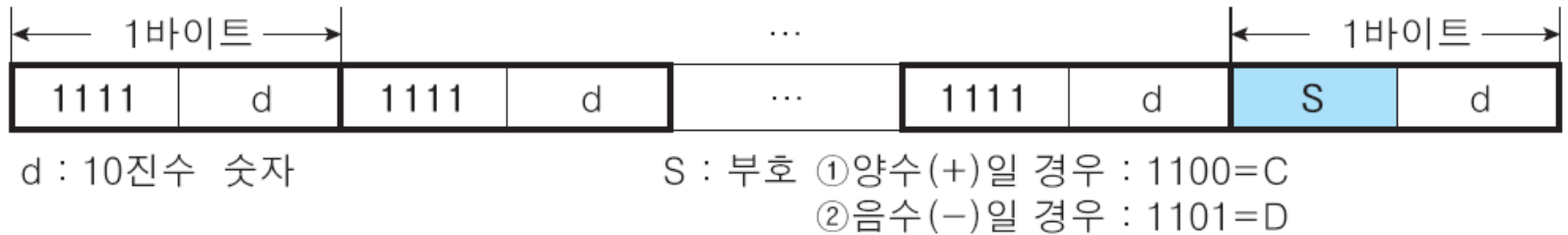
### ■ 4비트 수치 영역의 10진수 표현

수치 영역의 값				10진수 변환	10진수
0	0	0	0	$0 \times 8 + 0 \times 4 + 0 \times 2 + 0 \times 1$	0
0	0	0	1	$0 \times 8 + 0 \times 4 + 0 \times 2 + 1 \times 1$	1
0	0	1	0	$0 \times 8 + 0 \times 4 + 1 \times 2 + 0 \times 1$	2
0	0	1	1	$0 \times 8 + 0 \times 4 + 1 \times 2 + 1 \times 1$	3
0	1	0	0	$0 \times 8 + 1 \times 4 + 0 \times 2 + 0 \times 1$	4
0	1	0	1	$0 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1$	5
0	1	1	0	$0 \times 8 + 1 \times 4 + 1 \times 2 + 0 \times 1$	6
0	1	1	1	$0 \times 8 + 1 \times 4 + 1 \times 2 + 1 \times 1$	7
1	0	0	0	$1 \times 8 + 0 \times 4 + 0 \times 2 + 0 \times 1$	8
1	0	0	1	$1 \times 8 + 0 \times 4 + 0 \times 2 + 1 \times 1$	9
1	0	1	0	$1 \times 8 + 0 \times 4 + 1 \times 2 + 0 \times 1$	10 = A
1	0	1	1	$1 \times 8 + 0 \times 4 + 1 \times 2 + 1 \times 1$	11 = B
1	1	0	0	$1 \times 8 + 1 \times 4 + 0 \times 2 + 0 \times 1$	12 = C
1	1	0	1	$1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1$	13 = D
1	1	1	0	$1 \times 8 + 1 \times 4 + 1 \times 2 + 0 \times 1$	14 = E
1	1	1	1	$1 \times 8 + 1 \times 4 + 1 \times 2 + 1 \times 1$	15 = F



## □ 자료의 표현 - 10진수

- 여러 자리의 10진수의 존 형식 표현 방법
  - 10진수의 자릿수 만큼 존 형식을 연결하여 사용
  - 마지막 자리의 존 영역에 부호를 표시
    - 양수(+)는 1100
    - 음수(-)는 1101
  - 존 형식의 10진수 표현 형식



## □ 자료의 표현 - 10진수

- 존 형식으로 10진수를 표현하는 예

- +213

1111	0010	1111	0001	1100	0011
F	2	F	1	C(+)	3
2		1		3	

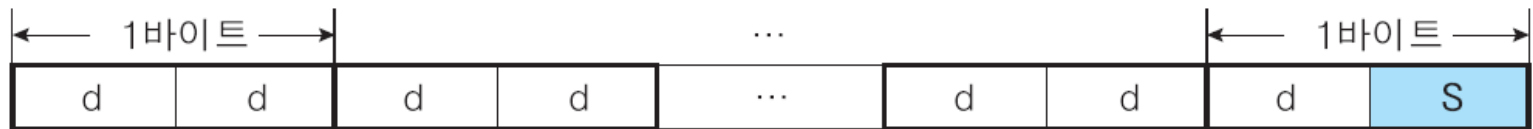
- -213

1111	0010	1111	0001	1101	0011
F	2	F	1	D(-)	3
2		1		3	

## □ 자료의 표현 - 10진수

### ❖ 팩 형식의 10진수(packed decimal)

- 10진수 한자리를 표현하기 위해 존 영역 없이 하나의 니블(4 bits)을 사용
  - 최하위 니블에 부호를 표시
    - 양수(+)는 1100
    - 음수(-)는 1101
- 여러 자리의 10진수의 팩 형식 표현 방법



d : 10진수 숫자

S : 부호 ①양수(+)일 경우 : 1100=C  
②음수(-)일 경우 : 1101=D

## □ 자료의 표현 - 10진수

- 팩 형식으로 10진수를 표현하는 예

- + 213

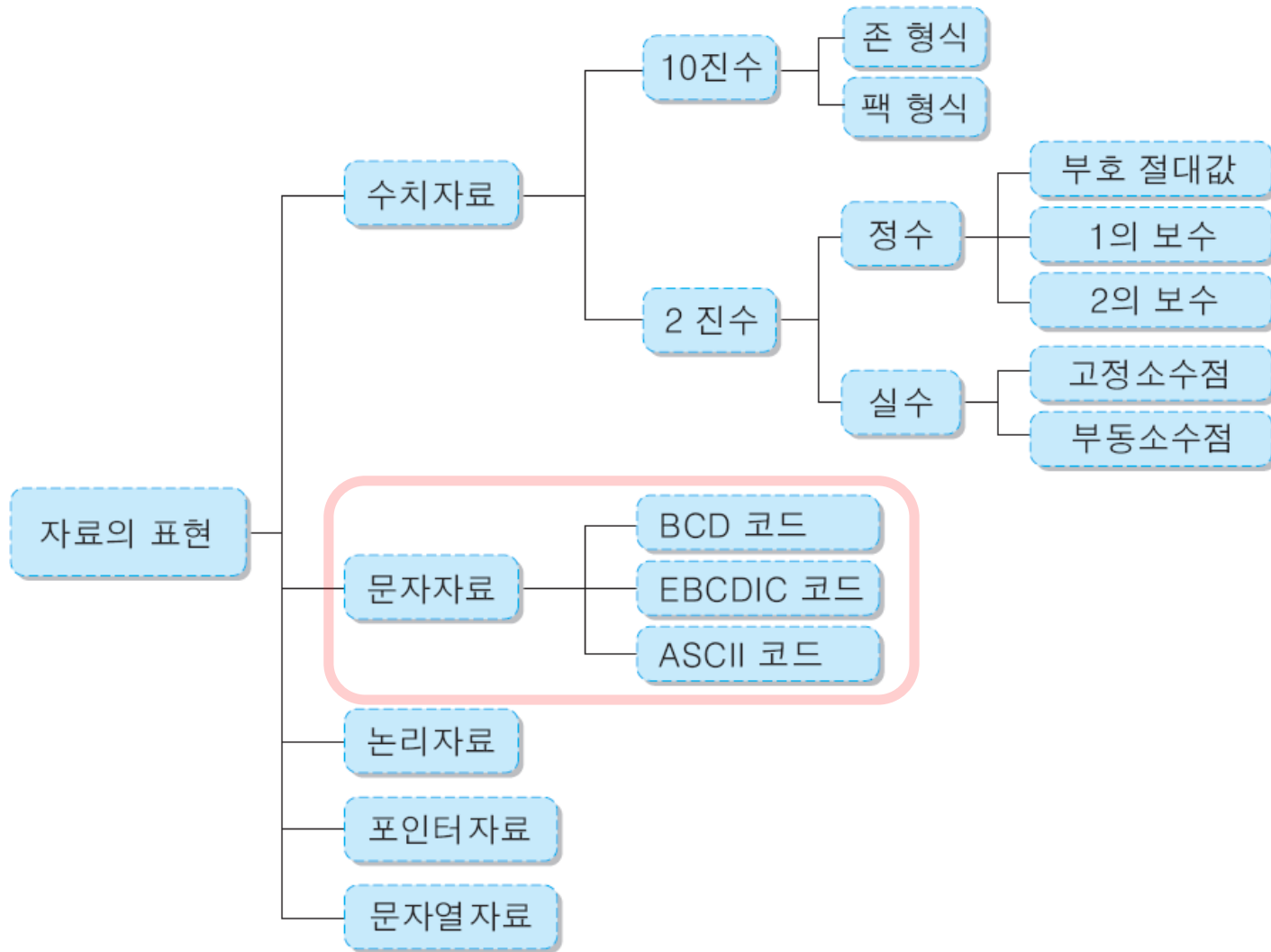
0010	0001	0011	1100
2	1	3	C(+)

- - 213

0010	0001	0011	1101
2	1	3	D(-)

## □ 자료의 표현

### ❖ 컴퓨터 내부에서 자료를 표현하는 방법



## □ 자료의 표현 - 문자 자료

### ❖ 문자자료의 표현

- 문자에 대한 이진수 코드를 정의하여 사용
  - 각 문자마다 서로 다른 값을 부여하기로 약속하고, 그 약속에 따라 특정 값이 특정 문자를 나타내는 것으로 본다.
- 여러 가지 문자 코드
  - BCD 코드
  - EBCDIC 코드
  - ASCII 코드 ➔ C 언어에서 사용
  - 유니코드(Unicode) ➔ 자바 언어에서 사용

## □ 자료의 표현 - 문자 자료

### ❖ ASCII 코드

- 7비트를 사용하여 문자 표현
  - 상위 3비트 : 존 비트
  - 하위 4비트 : 2진수 비트
  - 존 비트와 2진수 비트를 조합하여 10진수 0~9와 영어 대문자/소문자와 특수문자를 표현
- ASCII 코드의 구성

← 존 비트 →			← 숫자 비트 →			
			8	4	2	1
x	x	x	x	x	x	x

## □ 자료의 표현 - 문자 자료

### ■ ASCII 코드 표

- 예) 영문자 A에 대한 ASCII 코드 ➡ 1000001

하위 \ 상위	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	END	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FS	,	<	L	₩ (\)	l	
1101	CR	GS	-	=	M	]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

※ 코드의 의미

GS	Group Separator	RS	Record Separator	US	Unit Separator
----	-----------------	----	------------------	----	----------------



## □ 자료의 표현 - 문자 자료

### ❖ EBCDIC 코드

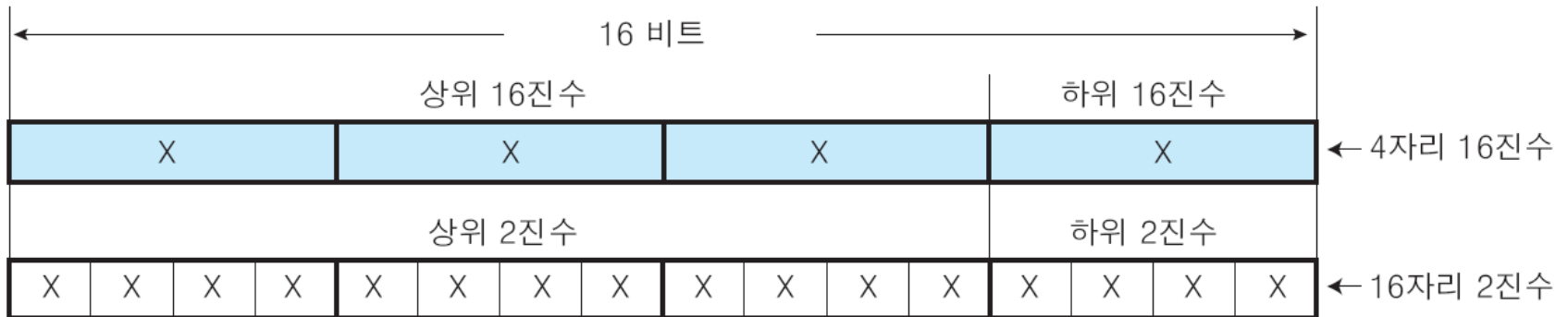
- 예) 영문자 A에 대한 EBCDIC 코드 ➡ 11000001

상위 하위	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	NUL	DLE	DS		SP	&	-						{	}	₩(\)	0
0001	SOH	DC1	SOS				/		a	j	~		A	J		1
0010	STX	DC2	FS	SYN					b	k	s		B	K	S	2
0011	ETX	TM							c	l	t		C	L	T	3
0100	PF	RES	BYP	PN					d	m	u		D	M	U	4
0101	HT	NL	LF	RS					e	n	v		E	N	V	5
0110	LC	BS	ETB	UC					f	o	w		F	O	W	6
0111	DEL	IL	ESC	EOT					g	p	x		G	P	X	7
1000	GE	CAN							h	q	y		H	Q	Y	8
1001	RLF	EM							i	r	z		I	R	Z	9
1010	SMM	CC	SM		∅	!		:								
1011	VT	CU1	CU2	CU3	.	\$	,	#								
1100	FF	IFS		DC4	<	*	%	@								
1101	CR	IGS	ENQ	NAK	(	)	-	'								
1110	SO	IRS	ACK		+	;	>	=								
1111	SI	IUS	BEL	SUB		┌	?	"								

## □ 자료의 표현 - 문자 자료

### ❖ 유니코드(Unicode)

- 나라별 언어를 모두 표현하기 위해 정의한 문자 코드
- 모든 문자를 16비트로 표현
  - 16비트의 코드값을 4자리의 16진수로 표시
- 유니코드의 구성



# □ 자료의 표현 - 문자 자료

## ■ 유니코드 표

상위16진수 하위16진수	000	001	002	003	004	005	006	007
0	NUL	DEL	SP	0	@	P	`	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	\$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	DEL

## □ 자료의 표현 - 문자 자료

- 영문자 'A'에 대한 유니코드

0041 ➔ 0000 0000 0100 0001



# □ 자료의 표현 - 문자 자료

## ■ 한글에 대한 유니코드

- 초성 19개 × 중성 21개 × 종성 28개 = 11,172개

[표 1-6] 유니코드표: 한글에 대한 유니코드 중 일부

상위16진수 하위16진수	AC0	AC1	AC2	AC3	AC4	AC5	AC6	AC7
0	가	감	갠	갬	갈	각	갯	거
1	각	갑	갬	갱	감	갈	갯	걱
2	갇	값	갬	갯	값	갯	갯	검
3	값	갇	갠	갯	값	갯	갯	겉
4	간	갇	갬	갬	갯	개	갬	건
5	값	강	갬	갬	갯	객	갬	겉
6	값	갇	갬	갯	값	객	갯	겉
7	간	갇	갬	갱	값	갯	갯	겉
8	갈	각	갯	가	감	갠	갯	겉
9	값	갈	갯	각	갑	갯	갱	겉
A	값	갯	갯	갇	값	갬	갯	겉
B	값	강	갬	갇	갇	갠	갯	겉
C	값	개	갬	간	갇	갬	객	겉
D	값	객	갬	값	강	갬	갬	겉
E	값	객	갯	값	갇	갬	갯	겉
F	값	갯	갯	갇	갇	갬	갱	겉

## □ 자료의 표현 - 문자 자료

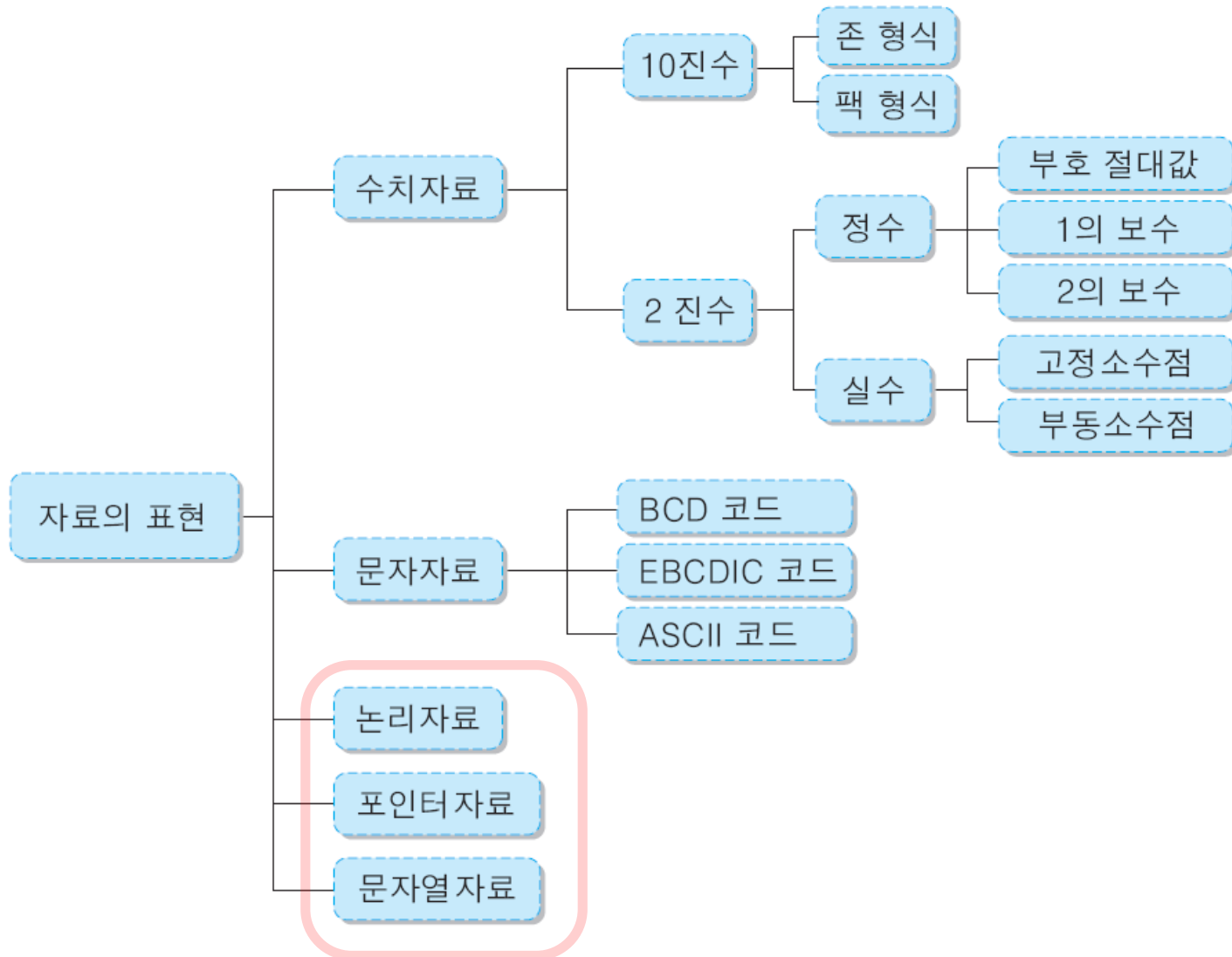
- 한글 '가'에 대한 유니코드

AC00 → 1010 1100 0000 0000

← 16 비트 →															
상위 16진수												하위 16진수			
A				C				0				0			
상위 2진수												하위 2진수			
1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0

## □ 자료의 표현

### ❖ 컴퓨터 내부에서 자료를 표현하는 방법



## □ 자료의 표현 - 논리 자료

### ❖ 논리 자료의 표현

- 참/거짓의 논리값을 표현하기 위한 자료 형식
- 1비트로도 표현 가능함
- 일반적으로 1바이트나 1워드를 사용하여 논리 자료를 표현
  - 방법1)
    - 참 - 최하위 비트를 1로 표시. 00000001
    - 거짓 - 전체 비트를 0으로 표시. 00000000
  - 방법2)
    - 참 - 전체 비트를 1로 표시. 11111111
    - 거짓 - 전체 비트를 0으로 표시. 00000000
  - 방법3)
    - 참 - 하나 이상의 비트를 1로 표시. 00000110
    - 거짓 - 전체 비트를 0으로 표시. 00000000



## □ 자료의 표현 - 포인터 자료

### ❖ 포인터(참조) 자료

- 메모리의 주소를 표현하기 위한 자료 형식
- 변수의 주소나 메모리의 특정 위치에 대한 주소를 저장하고 주소 연산하기 위해 사용

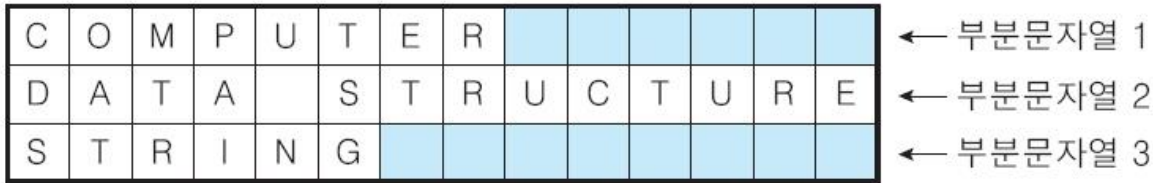
## □ 자료의 표현 - 문자열 자료

## ❖ 문자열(string) 자료의 표현

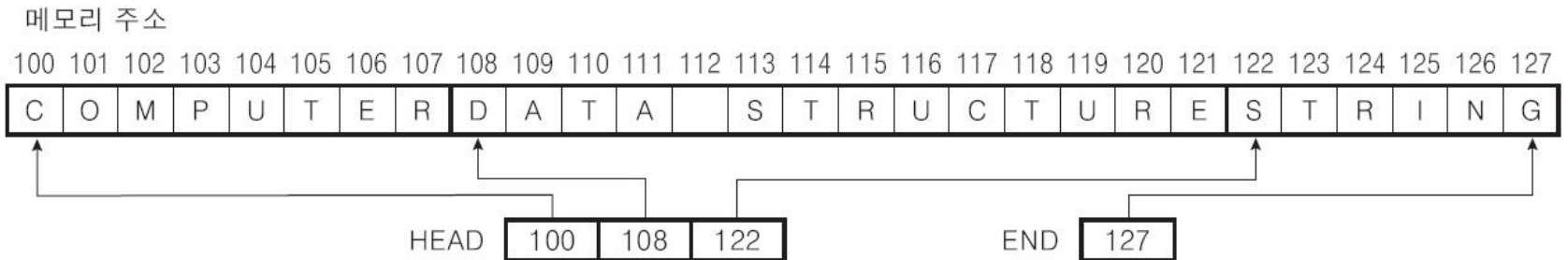
- 예) 표현할 문자열 : {COMPUTER, DATA STRUCTURE, STRING}
  - 방법1) 구분자를 사용하는 방법 : 구분자로 세미콜론(;) 사용한 예



- 방법2) 고정길이로 저장하는 방법



- 방법3) 포인터를 사용하는 방법



## □ 자료의 표현 - 문자열 자료

### ■ 문자열 표현 방법의 비교

비교항목 \ 방법	구분자를 사용하는 방법	고정길이로 저장하는 방법	포인터를 사용하는 방법
메모리 이용률	문자열 길이 + 구분자 길이 ☞ 효율적	가장 긴 부분문자열 길이 × 부분문자열의 개수 ☞ 비효율적	문자열 길이 + 포인터 저장공간 ☞ 효율적
부분문자열 탐색시간	문자 비교연산시간 + 구분자 식별시간 ☞ 비효율적	문자 비교연산시간 ☞ 효율적	문자 비교연산시간 + 포인터 주소연산시간 ☞ 효율적

예) 세 번째 부분문자열 STRING인가?