03 강

컴퓨터과학 개론

자료구조(1)

컴퓨터과학과 정광식 교수

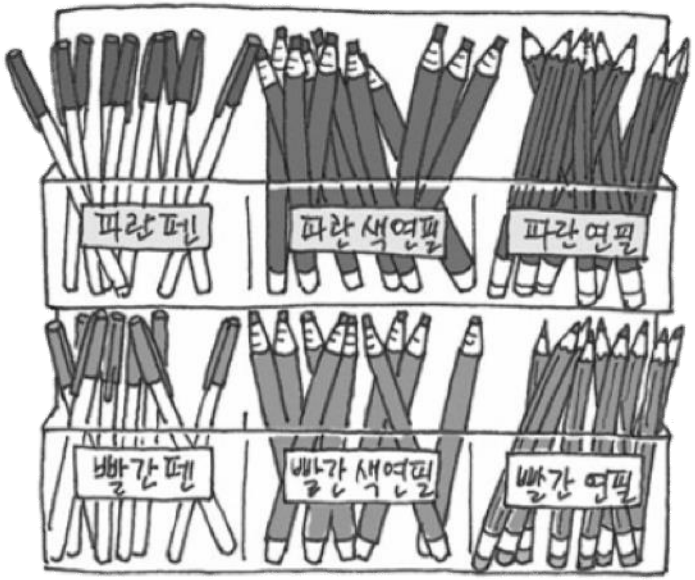
학습목차

01

기본 개념

자료구조의 개념

[나쁜 자료구조]



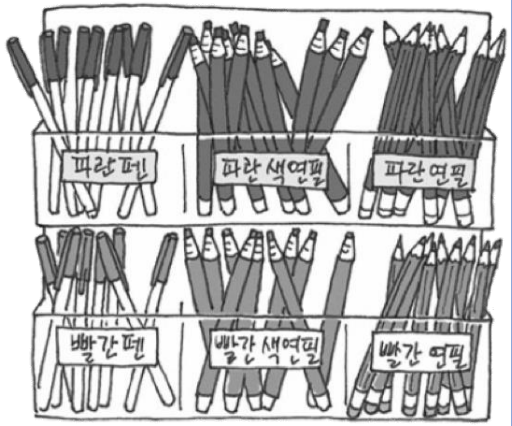
[좋은 자료구조]

http://www.incodom.kr/%EC%9E%90%EB%A3%8C%EA%B5%AC%EC%A1%B0

자료구조의 개념

기본개념

[나쁜 자료구조]



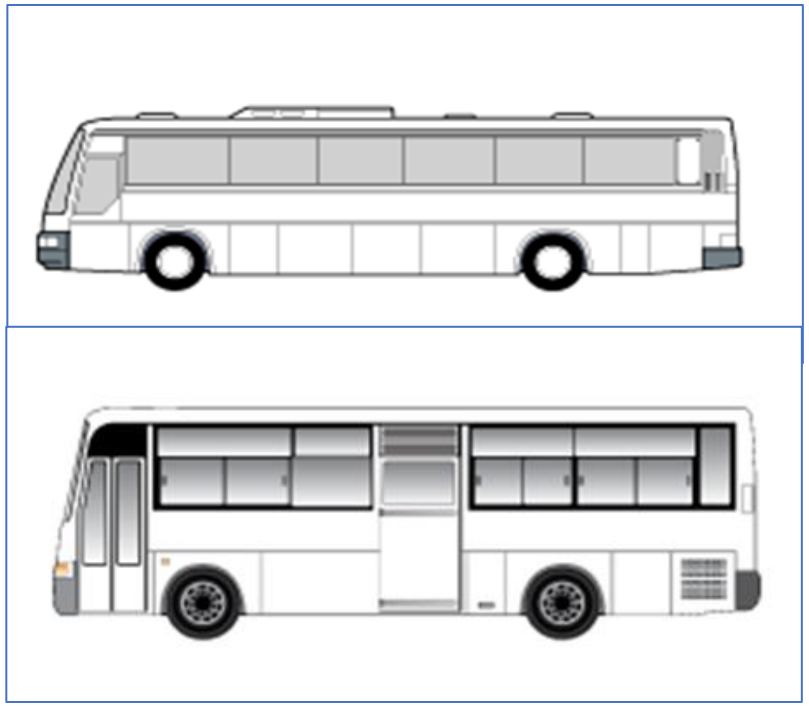
[좋은 자료구조]

	재고	사용량	구매예상 시기
파란펜	120	80	2019-09-01
빨간펜	110	90	2019-09-01
파란연필	190	10	2019-08-01
빨간연필	90	110	2019-10-01
총합	510	290	

http://www.incodom.kr/%EC%9E%90%EB%A3%8C%EA%B5%AC%EC%A1%B0

자료구조의 개념

추상화



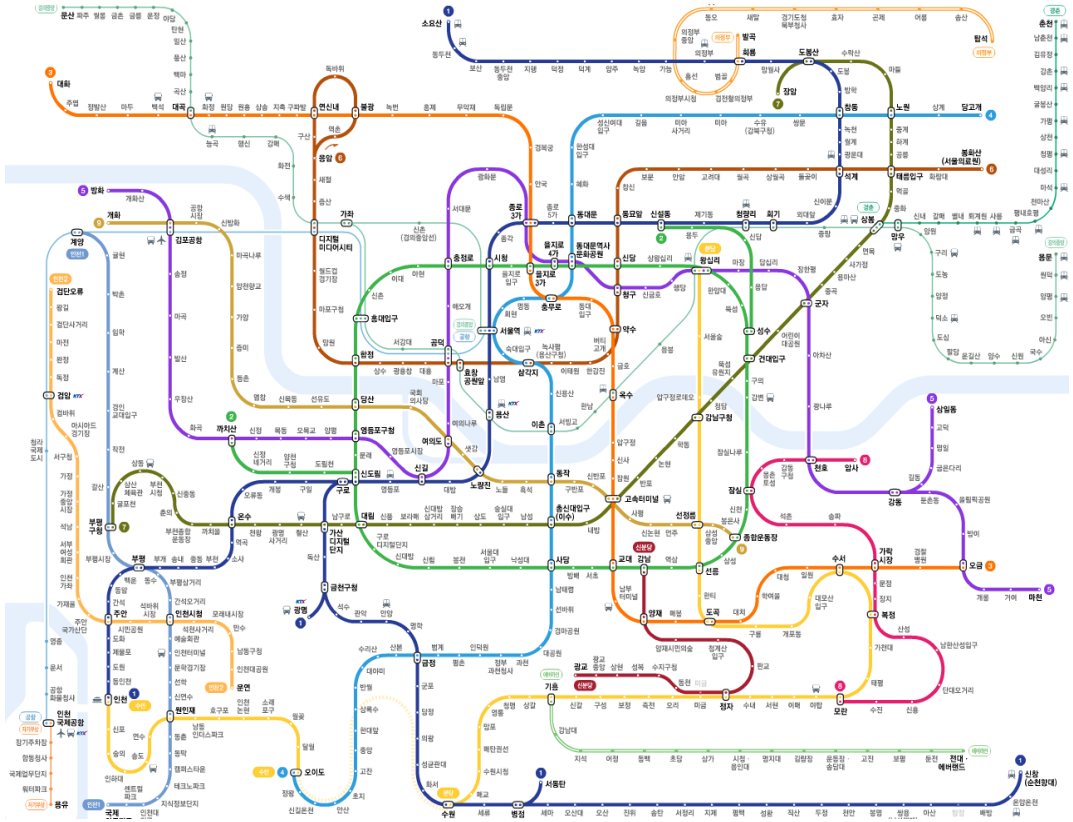
http://www.d-oz.net

3 + 2 = ?

5 * 4 = ?

자료구조의 개념

추상화



www.naver.com

자료구조의 개념

추상화와 구조화

- 자료의 추상화와 구조화가 적절히 이루어지지 못하면 소프트웨어는

  비효율적으로 개발되거나

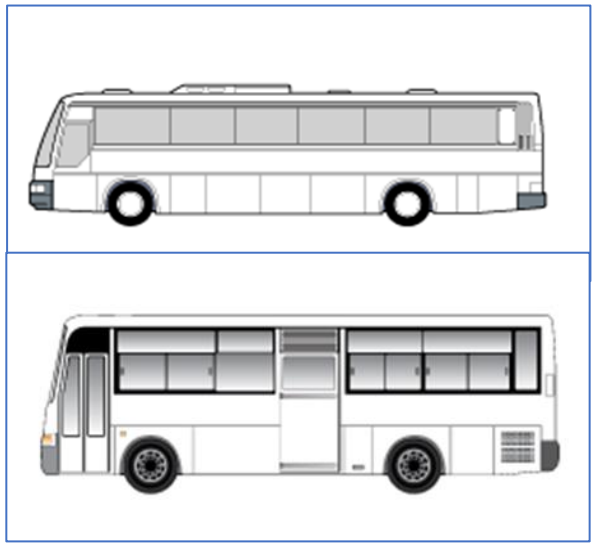
  비효율적으로 수행되거나

  소프트웨어의 확장성에 문제가 생기거나

  소프트웨어의 유지보수에 문제가 생기거나

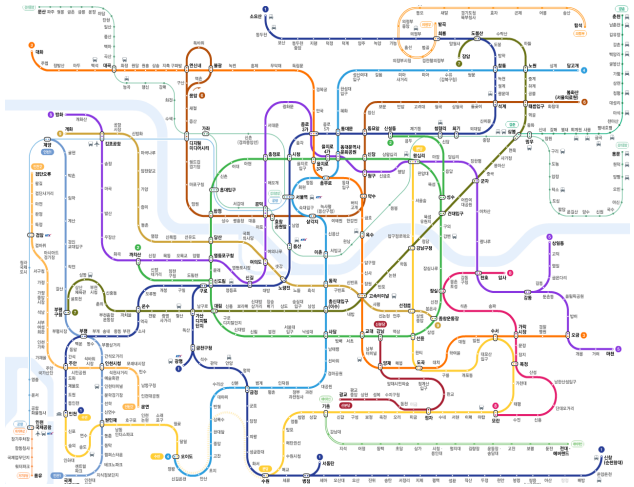
  할 수 있음

자료구조의 개념



3 + 2 = ?

5 * 4 = ?



자료구조의 개념

추상화

- 공통적인 개념을 이용하여 같은 종류의 다양한 객체를 정의하는 것
  - 수식, 프로그램 언어 등

자료(데이터) 추상화

- 다양한 객체를 컴퓨터에서 표현하고 활용하기 위해 필요한 데이터의 구조에 대해서 공통의 특징만을 뽑아 정의한 것

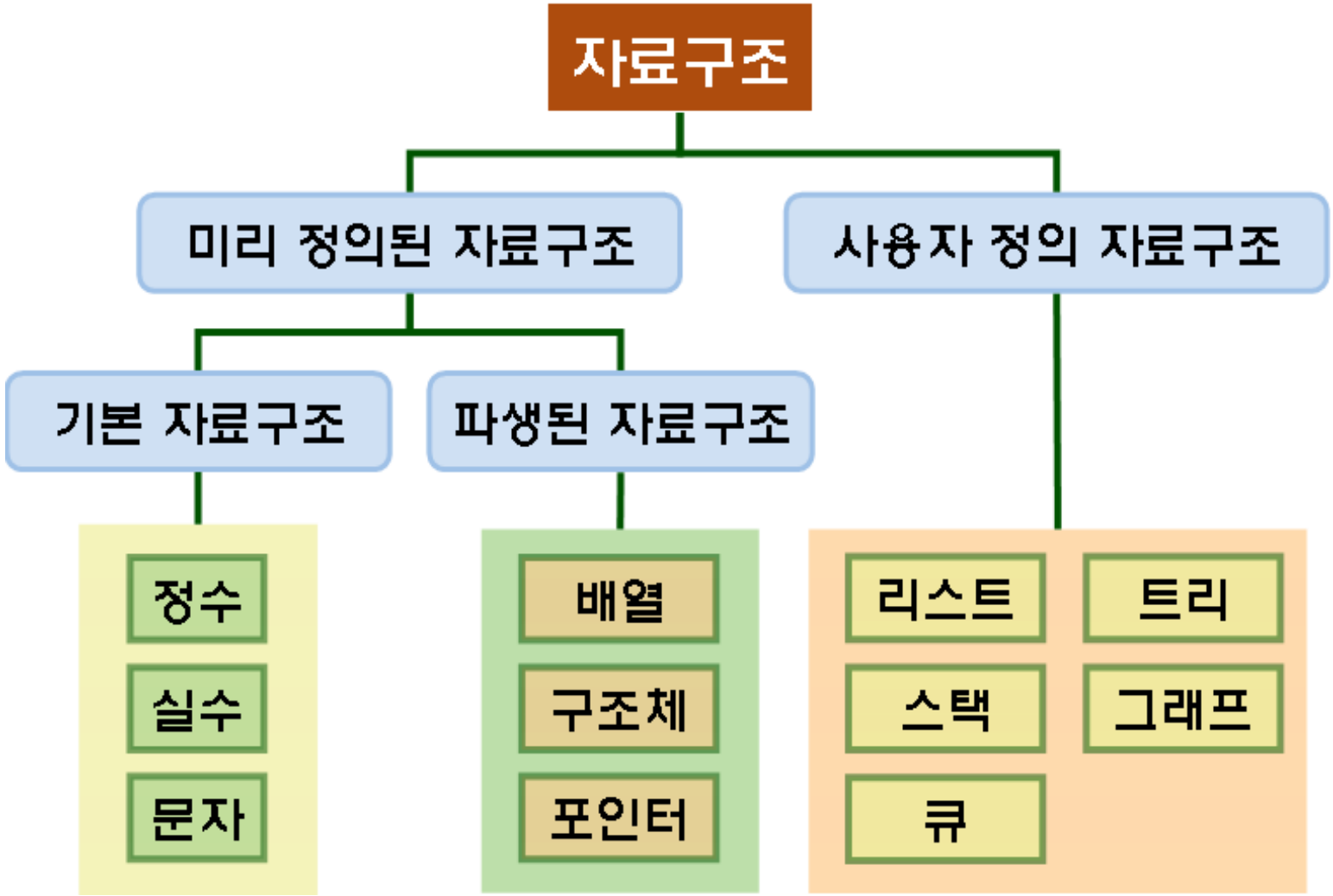
⇒ 왜 ????

자료구조의 개념

자료의 추상화

- 자료 사이의 논리적 관계를 컴퓨터나 프로그램에 적용하기 위해서는 자료의 추상화가 필요함
  - 자료구조(data structure) : 추상화를 통해 자료의 논리적 관계를 구조화한 것
- 자료가 복잡해지거나 소프트웨어가 복잡해 질수록 자료구조의 중요성이 강조됨

자료구조의 종류와 관계


자료구조
미리 정의된 자료구조
사용자 정의 자료구조
기본 자료구조
파생된 자료구조
정수
실수
문자
배열
구조체
포인터
리스트
트리
스택
그래프
큐

자료구조의 개념

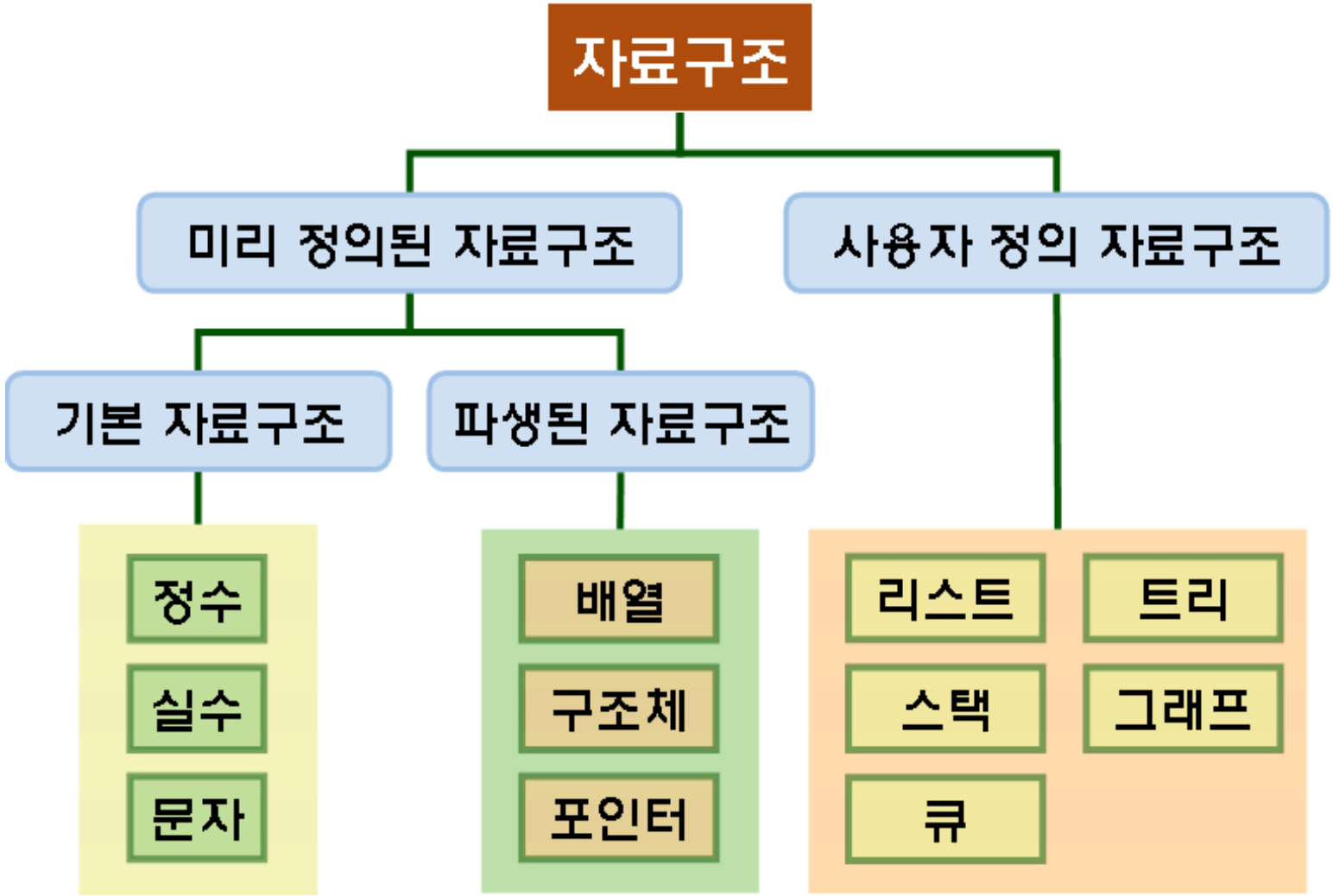
미리 정의된 자료구조

- 프로그래밍 언어에서 제공함
- 프로그래밍 설계나 컴파일러 구현 단계에서 정의되어 개발자에게 제공되는 자료구조

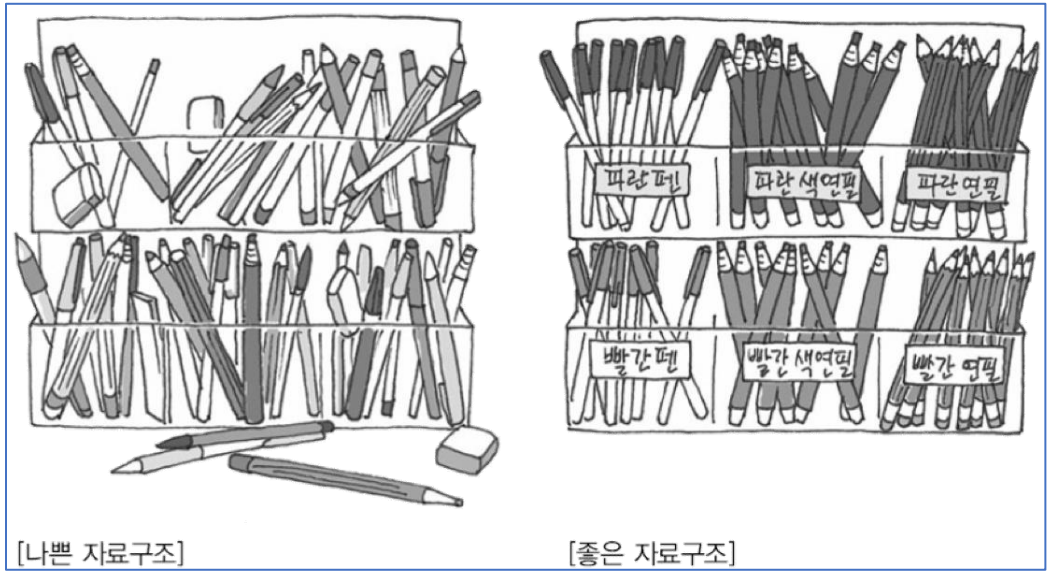
사용자 정의 자료구조

- 개발자가 정의하여 사용함
- 소프트웨어 개발 중에 개발자에 의해 만들어지는 자료구조 (리스트, 스택, 큐, 트리, 그래프 등)

자료구조의 종류와 관계


자료구조
미리 정의된 자료구조
사용자 정의 자료구조
기본 자료구조
파생된 자료구조
정수
실수
문자
배열
구조체
포인터
리스트
트리
스택
그래프
큐

자료구조의 개념



[나쁜 자료구조] [좋은 자료구조]

	재고	사용량	구매예상 시기
파란펜	120	80	2019-09-01
빨간펜	110	90	2019-09-01
파란연필	190	10	2019-08-01
빨간연필	90	110	2019-10-01
총합	510	290	

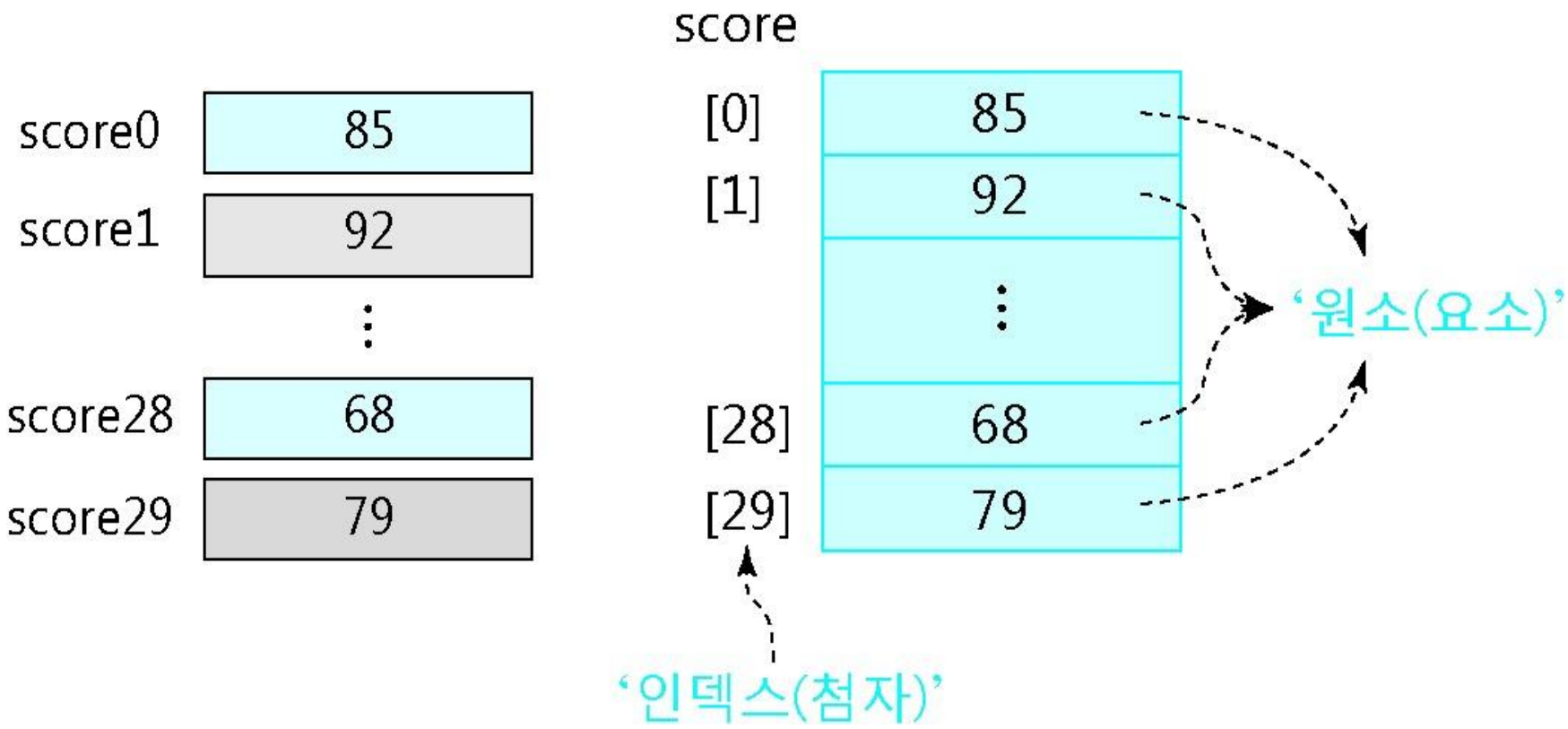
http://www.incodom.kr/%EC%9E%90%EB%A3%8C%EA%B5%AC%EC%A1%B0

02

배열

배열의 개념

배열



(a) 다수의 개별 변수의 사용

(b) 배열의 사용

배열의 개념

배열(array)

- 동일한 자료형을 갖는 여러 개의 데이터를 동일한 변수 이름의 방에 일렬로 저장하는 자료 집합체 (원소+인덱스)

원소(요소)

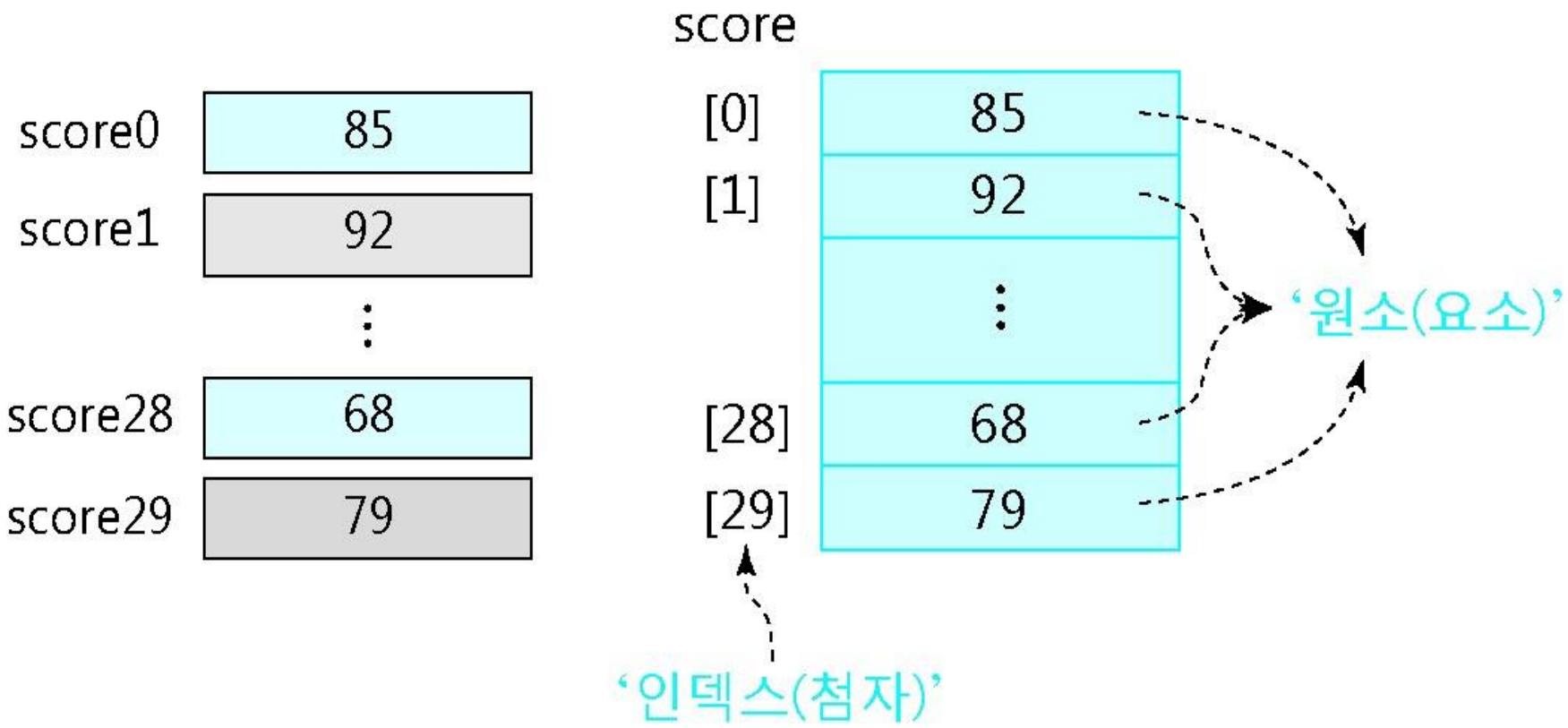
- 자료 집합체에서 각 원소의 항목값 : 데이터

인덱스(첨자)

- 자료 집합체에서 각 원소가 저장된 방을 접근하기 위한 방 번호에 해당하는 것 : 번호

배열의 개념

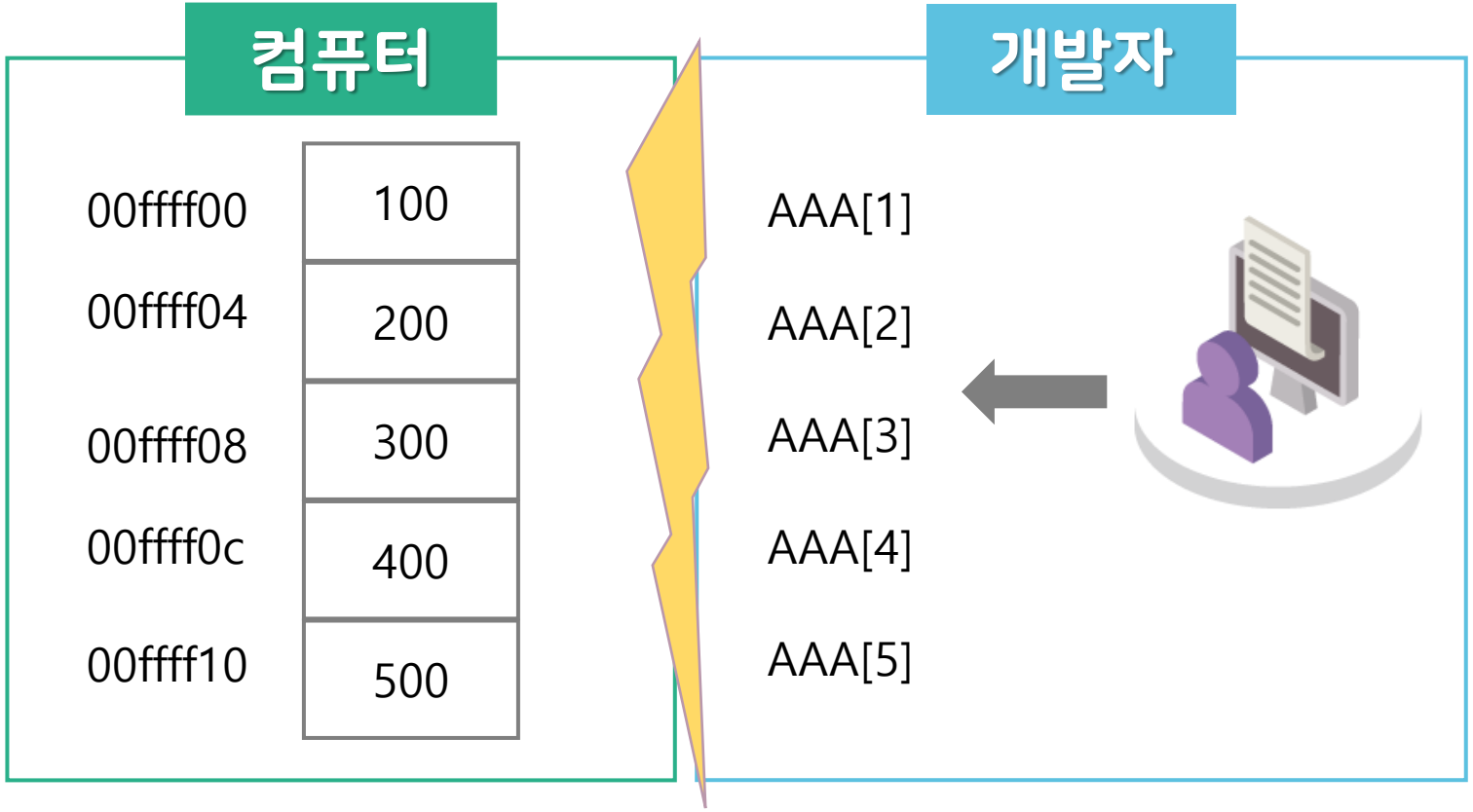
배열



(a) 다수의 개별 변수의 사용

(b) 배열의 사용

1차원 배열에서의 주소 계산

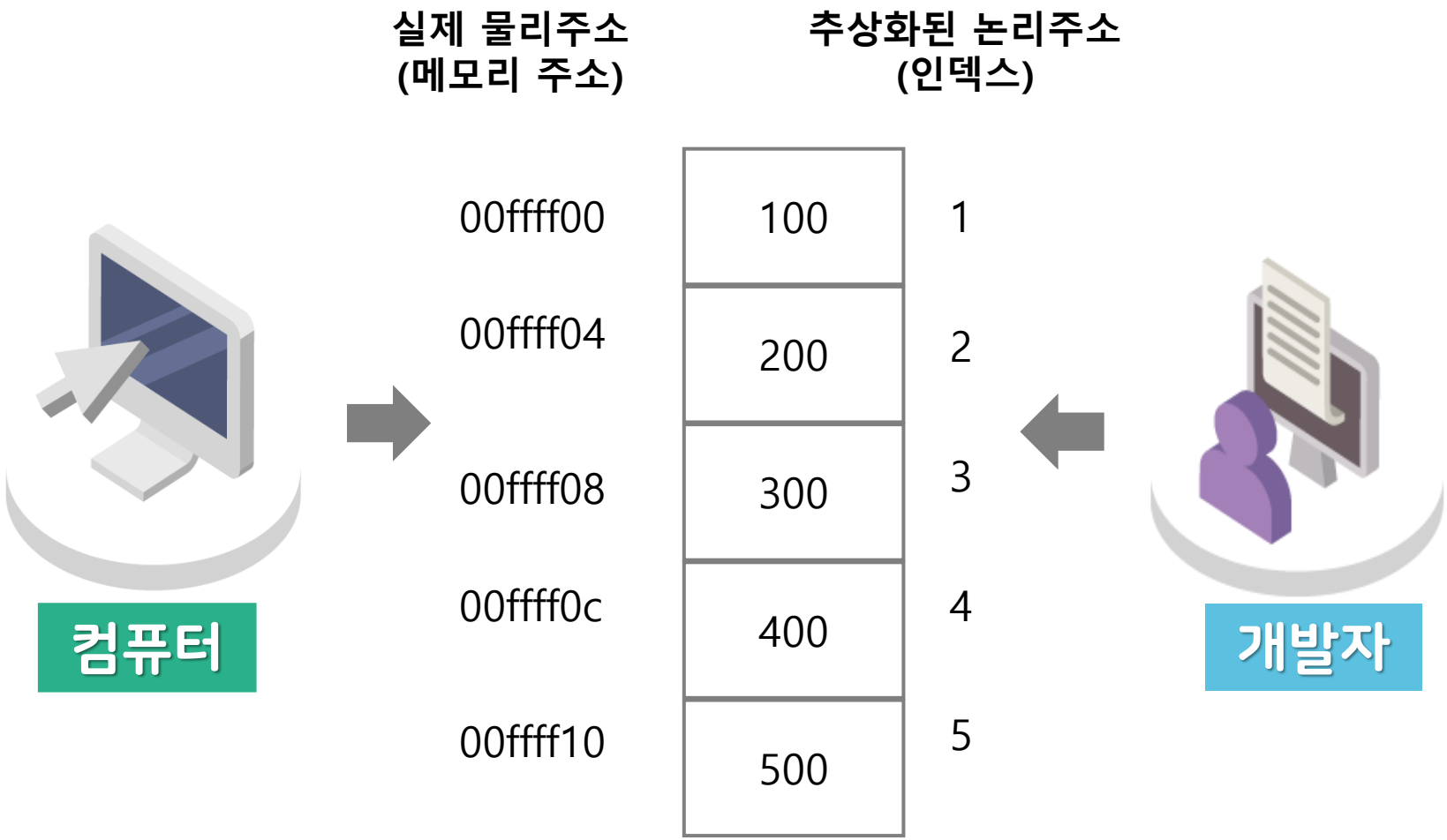


1차원 배열

개념

- 가장 간단한 형태의 배열임
- 한 개의 인덱스(첨자)를 사용해서 원소에 직접 접근함
- 배열의 원소들은 컴퓨터 메모리의 연속적인 기억장소에 할당되어 순차적으로 저장됨
- 배열 A의 크기를 k라고 가정하고 시작 주소를 a 라고 가정하면, A[i]의 저장 주소는 $a+l*k$ 가 됨

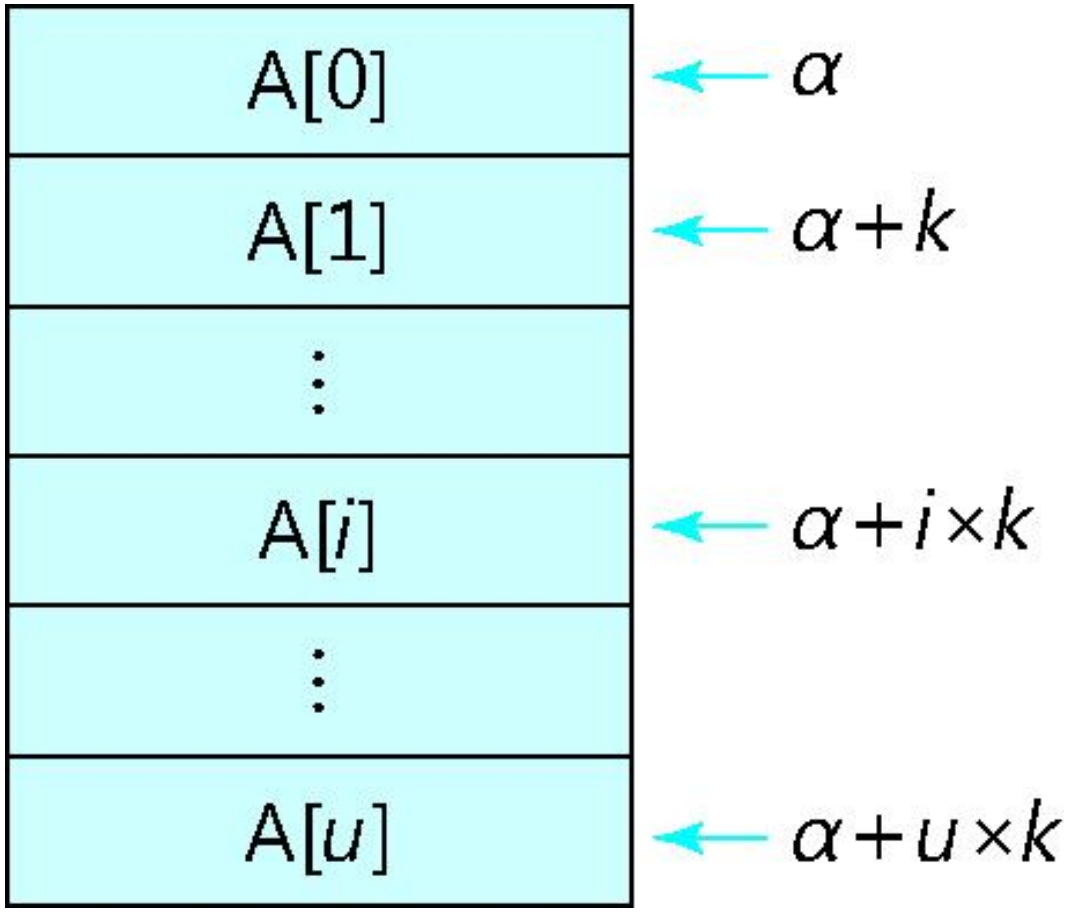
1차원 배열에서의 주소 계산



1차원 배열에서의 주소 계산

컴퓨터

00ffff00	100
00ffff04	200
00ffff08	300
00ffff0c	400
00ffff10	500



다차원 배열

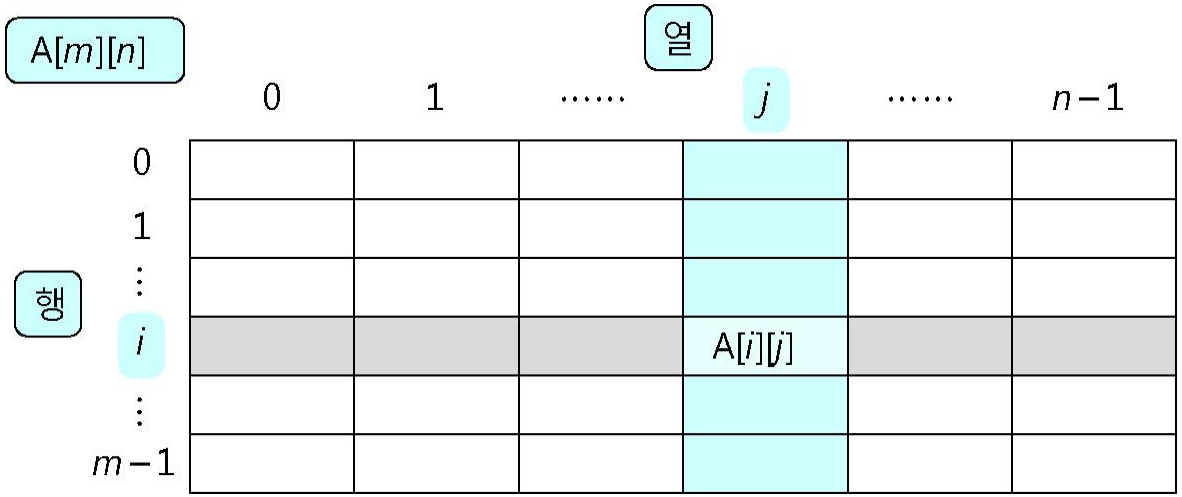
2차원 배열

- 두개의 첨자를 가지는 배열
- 동일한 크기의 1차원 배열을 모아 놓아, 바둑판 형태로 만든 배열
- 하나의 원소는 두 개의 첨자 i 와 j 의 쌍으로 구분됨 : A[i][j]

다차원 배열

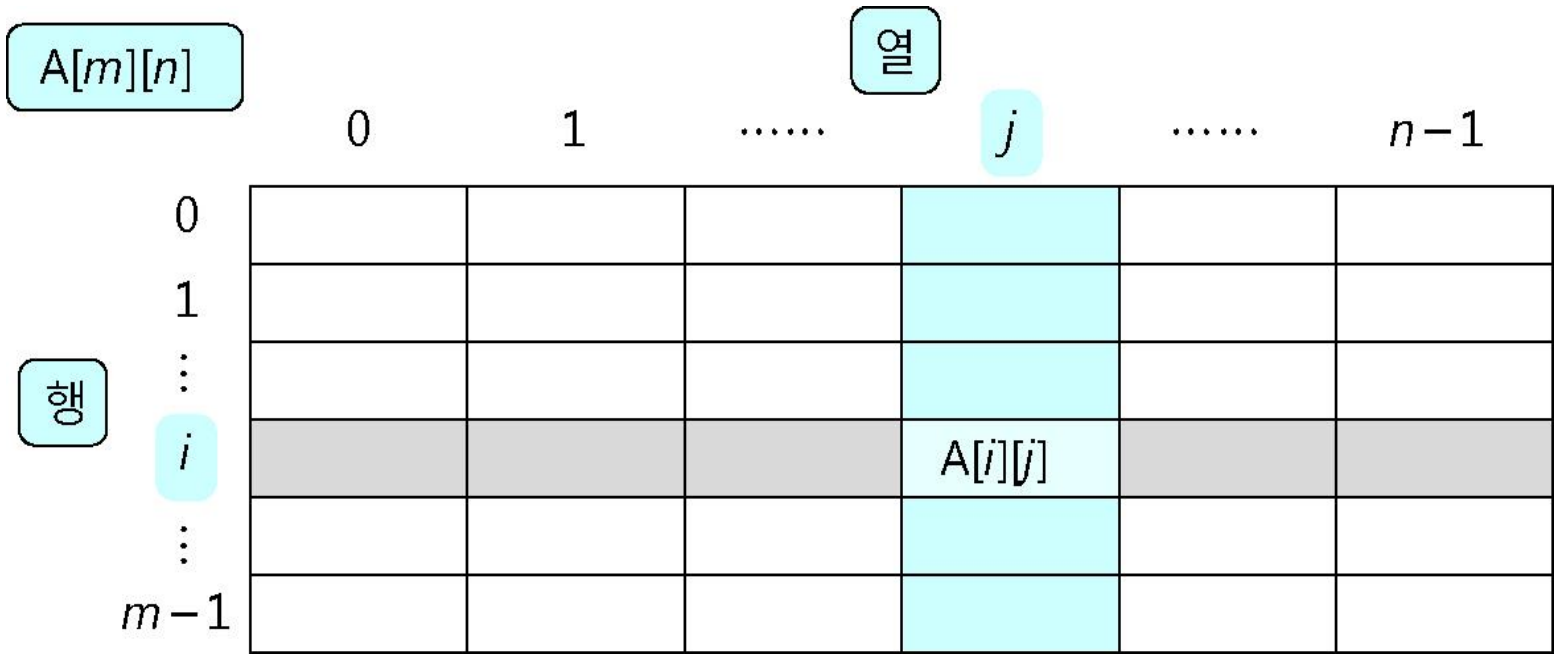
2차원 배열

- 행(row) : 첨자 i 에 해당하는 것
- 열(column) : 첨자 j 에 해당하는 것



다차원 배열

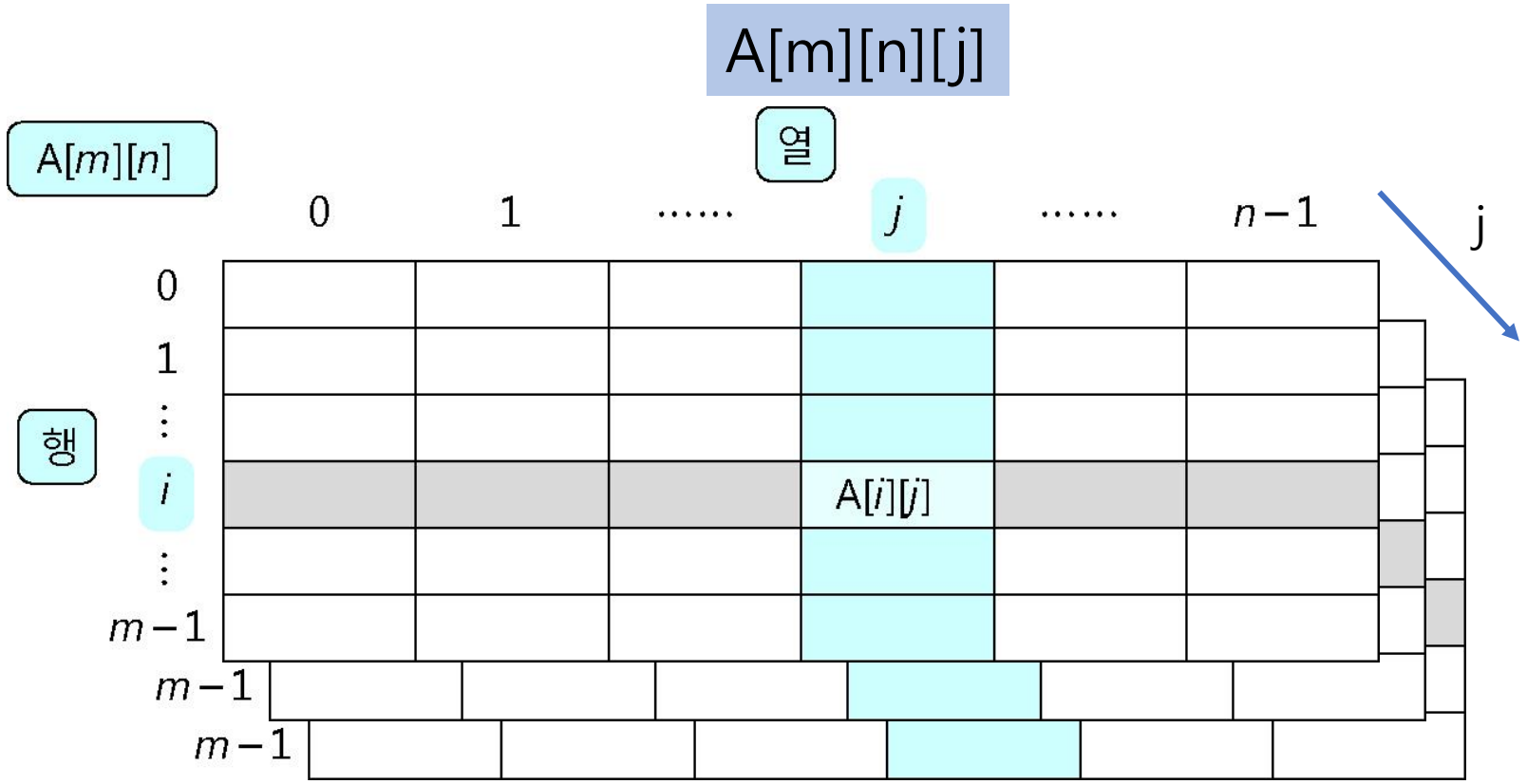
2차원 배열



다차원 배열

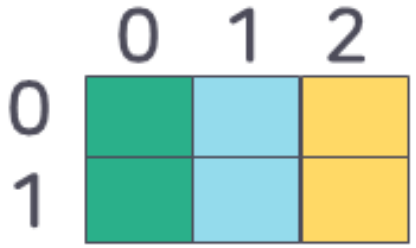
3차원 배열

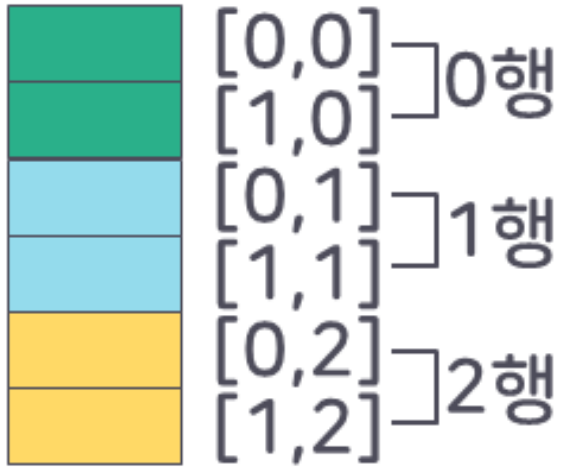
- 세개의 첨자들을 가지는 배열



2차원 배열 저장 순서

열 우선 순서 저장





2차원 배열 저장 순서

열 우선 순서 저장

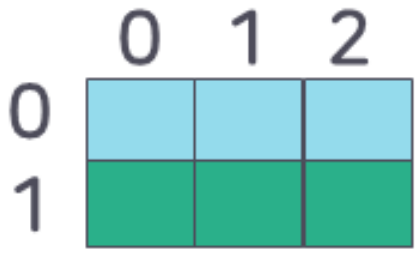
- 첫 열에 있는 각 행의 원소를 차례대로 컴퓨터 메모리에 저장하고 다음 열로 이동하여 각 행에 있는 원소를 차례대로 컴퓨터 메모리에 저장하는 방법

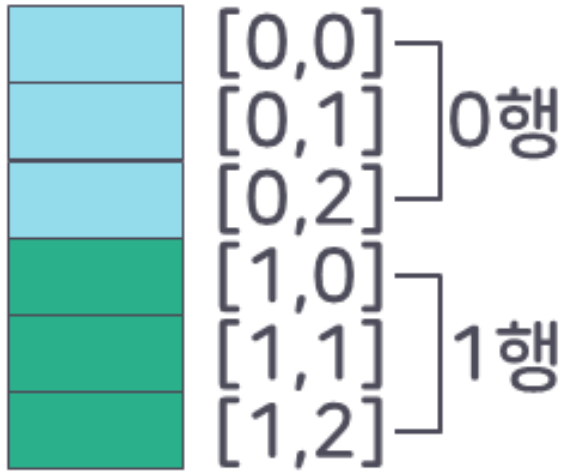
2차원 배열 저장 순서

열 우선 순서 저장

2차원 배열 저장 순서

행 우선 순서 저장





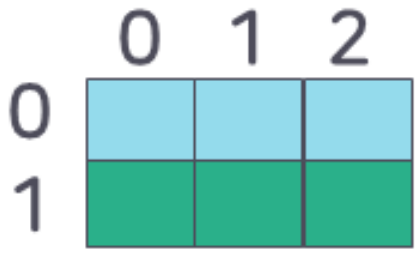
2차원 배열 저장 순서

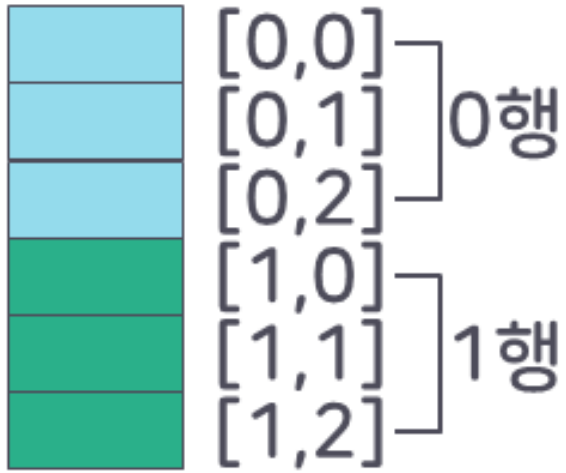
행 우선 순서 저장

- 첫 행에 있는 각 열의 원소를 차례대로 컴퓨터 메모리에 저장하고 다음 행으로 이동하여 각 열에 있는 원소부터 차례대로 컴퓨터 메모리에 저장하는 방법

2차원 배열 저장 순서

행 우선 순서 저장





희소 행렬(spare matrix)

$$A = \begin{pmatrix} 0 & 20 & 0 & 0 & 9 & 0 & 0 & 11 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 78 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 67 & 0 & 0 & 0 & 0 \\ 0 & 31 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 91 & 0 & 0 & 44 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 19 & 0 & 0 & 27 & 0 \end{pmatrix}$$

희소 행렬(spare matrix)

개념

- 원소 값이 0 인 원소가 그렇지 않은 원소보다 상대적으로 많은 행렬
- 0 값을 저장하기 위해 컴퓨터 메모리의 낭비를 막고, 처리의 효율성을 높이기 위해 사용됨
- 희소 행렬의 0 인 원소는 저장하지 않고, 0 이 아닌 값 만을 따로 모아서 저장하는 방법
- 0 이 아닌 각 원소를 (행 번호, 열 번호, 원소 값)의 형태로 나타내면, 2차원 배열로 표현 가능함

희소 행렬의 일반적인 2차원 배열 표현

0	20	0	0	9	0	0	11	0
0	0	0	0	0	0	0	0	0
78	0	0	0	0	0	0	0	0
0	0	0	0	67	0	0	0	0
0	31	0	0	0	0	0	0	0
0	0	0	91	0	0	44	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

$$A = \begin{pmatrix} 0 & 20 & 0 & 0 & 9 & 0 & 0 & 11 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 78 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 67 & 0 & 0 & 0 & 0 \\ 0 & 31 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 91 & 0 & 0 & 44 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 19 & 0 & 0 & 27 & 0 \end{pmatrix}$$

희소 행렬의 효율적 표현

$$A = \begin{pmatrix} 0 & 20 & 0 & 0 & 9 & 0 & 0 & 11 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 78 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 67 & 0 & 0 & 0 & 0 \\ 0 & 31 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 91 & 0 & 0 & 44 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 19 & 0 & 0 & 27 & 0 \end{pmatrix}$$

	행	열	값
0	8	9	10
1	0	1	20
2	0	4	9
3	0	7	11
4	2	0	78
5	3	4	67
6	4	1	31
7	5	3	91
8	5	6	44
9	7	4	19
10	7	7	27

희소 행렬의 효율적 표현

0	20	0	0	9	0	0	11	0
0	0	0	0	0	0	0	0	0
78	0	0	0	0	0	0	0	0
0	0	0	0	67	0	0	0	0
0	31	0	0	0	0	0	0	0
0	0	0	91	0	0	44	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

8	9	10
0	1	20
0	4	9
0	7	11
2	0	78
3	4	67
4	1	31
5	3	91
5	6	44
7	4	19
7	7	27

03

리스트

선형 리스트(linear list)

개념(1)

- 순서 리스트(ordered list)라고도 함
- 1개 이상의 원소들이 순서를 가지고 구성됨
- $A = (a_1, a_2, \cdots, a_i, \cdots, a_n)$ 과 같이 표시하며,

  a_i 는 i 번째 원소를 나타내고, a_n 의 n 은 리스트의 크기가 됨

선형 리스트(linear list)

개념(2)

- 요일 리스트 : (월, 화, 수, 목, 금, 토, 일)

- 전쟁 리스트: ((황산벌 전투, 660), (임진왜란, 1592),
  (세계 1차 대전, 1914), (세계 2차 대전, 1939))

선형 리스트의 구현(배열)

개념

- 선형 리스트와 1차원 배열은 순차적인 구조를 가지고 있으므로

  1차원 배열로 간단하게 표현할 수 있음

선형 리스트의 구현(배열)

삽입

- 원소를 삽입하기 위해서는 삽입될 위치 이후의 원소들의 순서를 그대로 유지하면서 원소를 삽입해야 함
- 삽입할 위치에 있는 원소와 그 다음의 원소들을 모두 한 칸 씩 뒤로 이동시켜야 함

선형 리스트의 구현(배열)

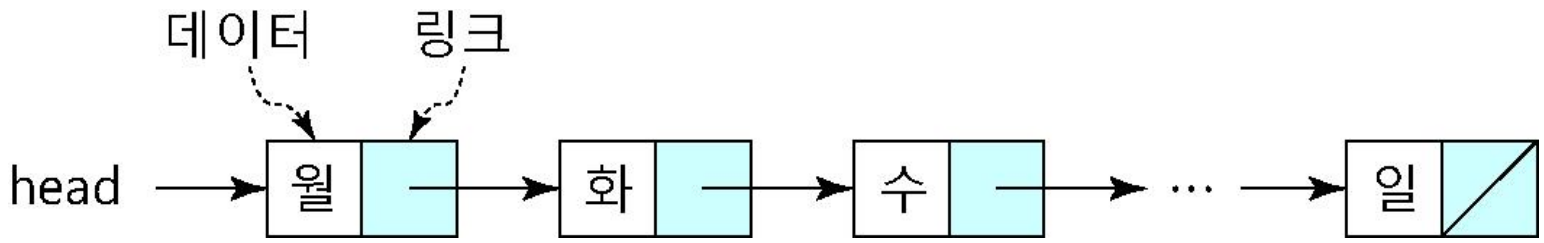
삭제

- 원소 삭제의 경우에도 삭제할 원소를 찾아 삭제한 후, 그 뒤에 있는 모든 원소들을 한 칸 씩 앞으로 이동시켜야 함

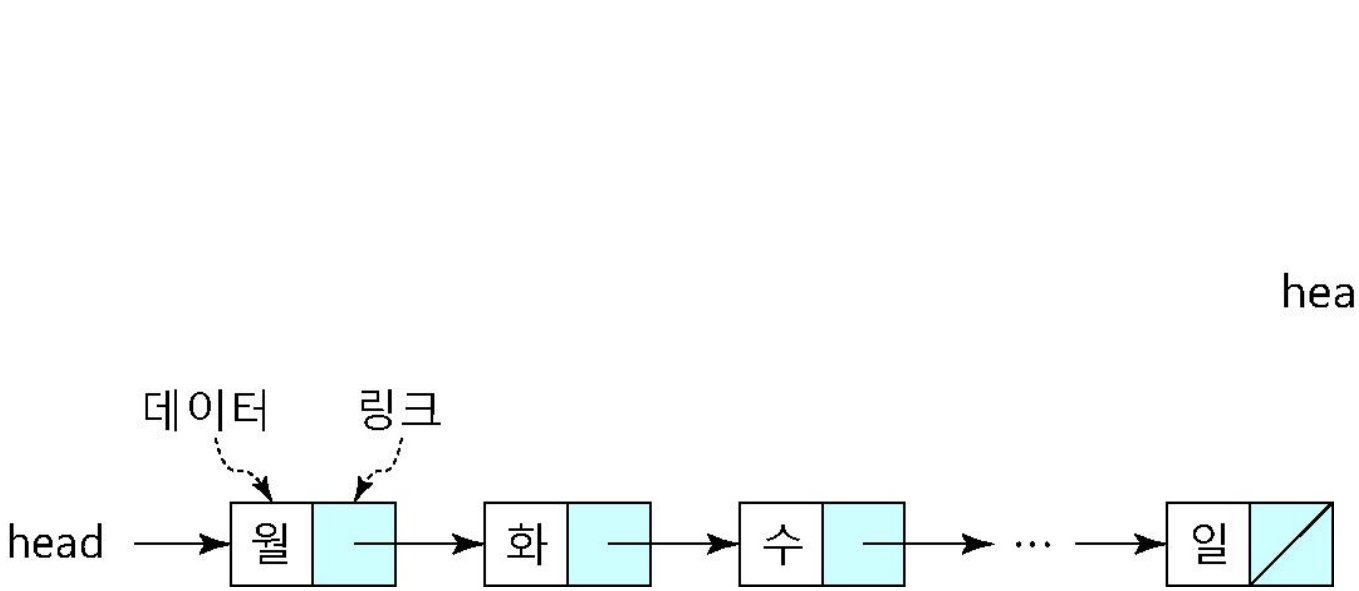
선형 리스트의 구현(연결 리스트)

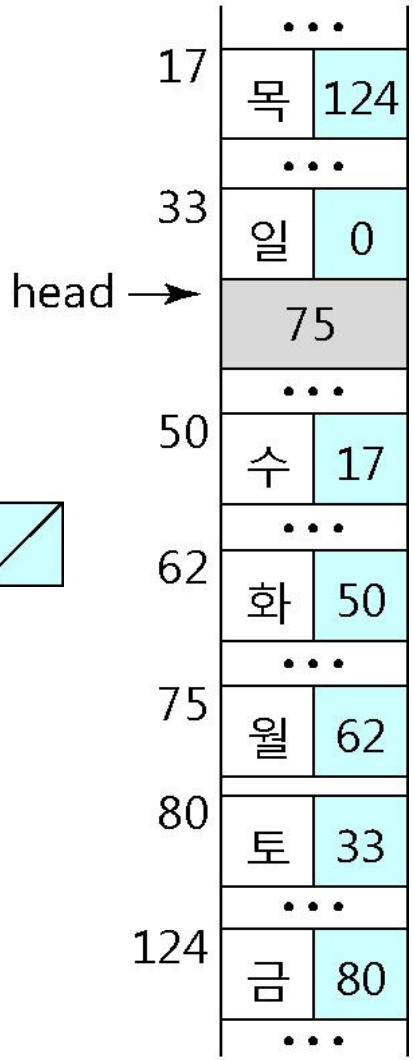
개념

- 노드 간의 포인터 연결을 통해서 구현됨
- 각 노드는 적어도 두 종류의 필드, 원소 값을 저장하는 데이터 필드와 노드 연결을 위한 링크 필드를 가짐
- 선형 리스트의 논리적 순서만을 지원함



선형 리스트의 구현(연결 리스트)





연결 리스트 종류

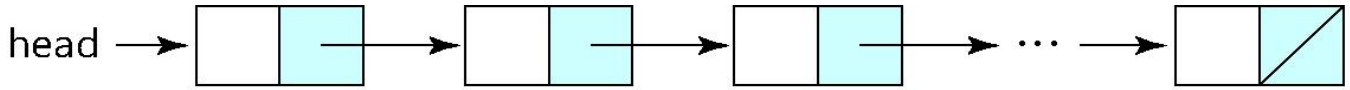
단일 연결 리스트(singly linked list)

- 특정 노드의 링크 필드를 사용해서 후행 노드를 가리킴
- 특정 노드의 후행 노드는 쉽게 접근할 수 있지만,

  선행 노드에 대한 접근은 헤드 노드부터 새로 시작해야 함

연결 리스트 종류

단일 연결 리스트(singly linked list)



(a) 단일 연결 리스트

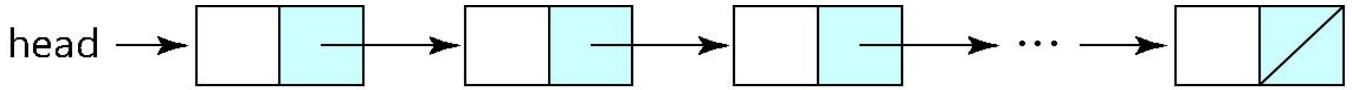
연결 리스트 종류

이중 연결 리스트(doubly linked list)

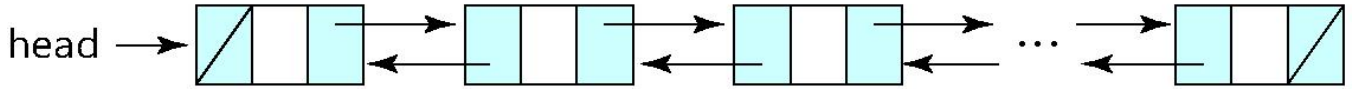
- 특정 노드의 첫번째 링크는 후행 노드를 가리키고
  두번째 링크는 선행 노드를 가리킴
- 특정 노드에서 후행 노드 뿐만 아니라 선행 노드에 대한 접근을 쉽게 제공하기 위한 것

연결 리스트 종류

이중 연결 리스트(doubly linked list)



(a) 단일 연결 리스트

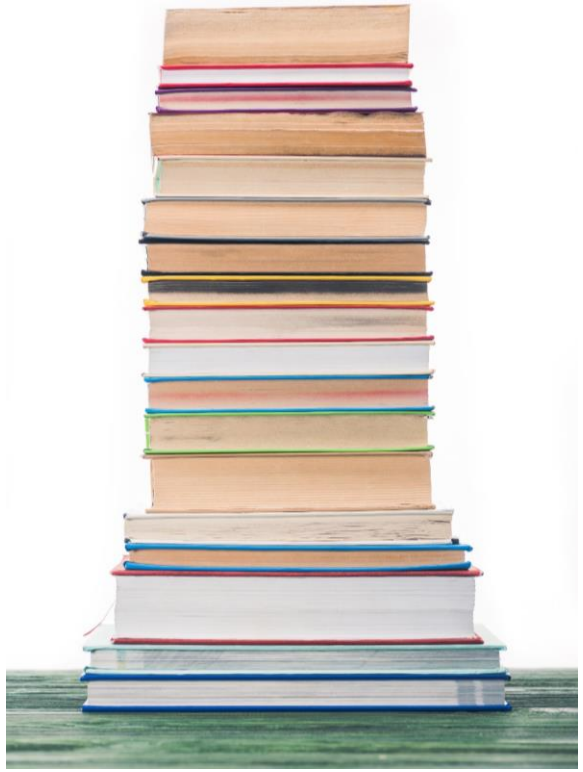


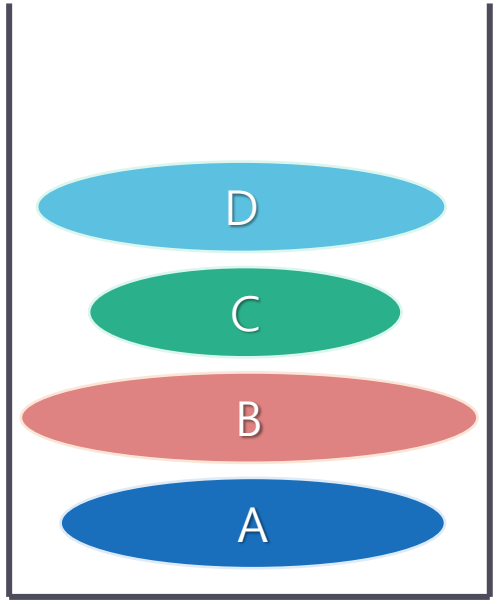
(b) 이중 연결 리스트

04

스택과 큐

스택(Stack)

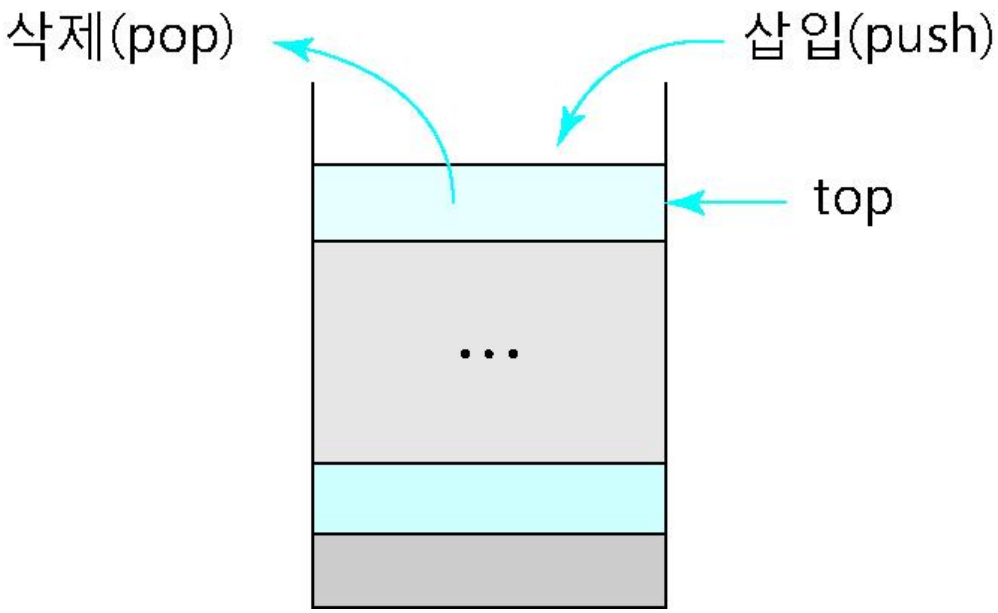




https://www.freepik.com/premium-photo/stack-books-isolated_18591034.htm

https://create.vista.com/unlimited/stock-photos/190885422/stock-photo-high-tower-stacked-books-table/

스택(Stack)



스택(Stack)

개념

- 데이터의 삽입과 삭제가 한쪽 끝에서만 이루어지는 자료구조
- 가장 먼저 입력된 데이터가 가장 나중에 제거되는 선입후출(FILO, First-In-Last-out) 특징을 가짐

스택의 연산

스택 오버플로 (overflow)

- 삽입 연산을 수행할 때 발생함
- 스택을 위해 할당된 저장 공간을 초과해서 더 이상 데이터를 삽입할 수 없는 현상

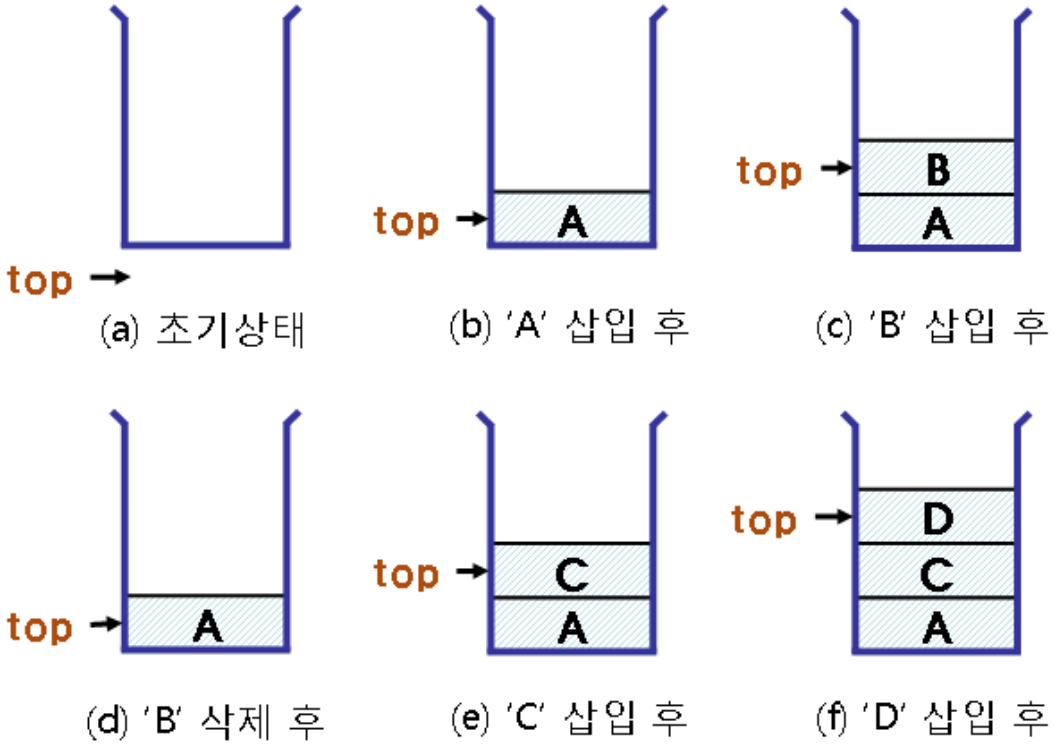
스택의 연산

스택 언더플로 (underflow)

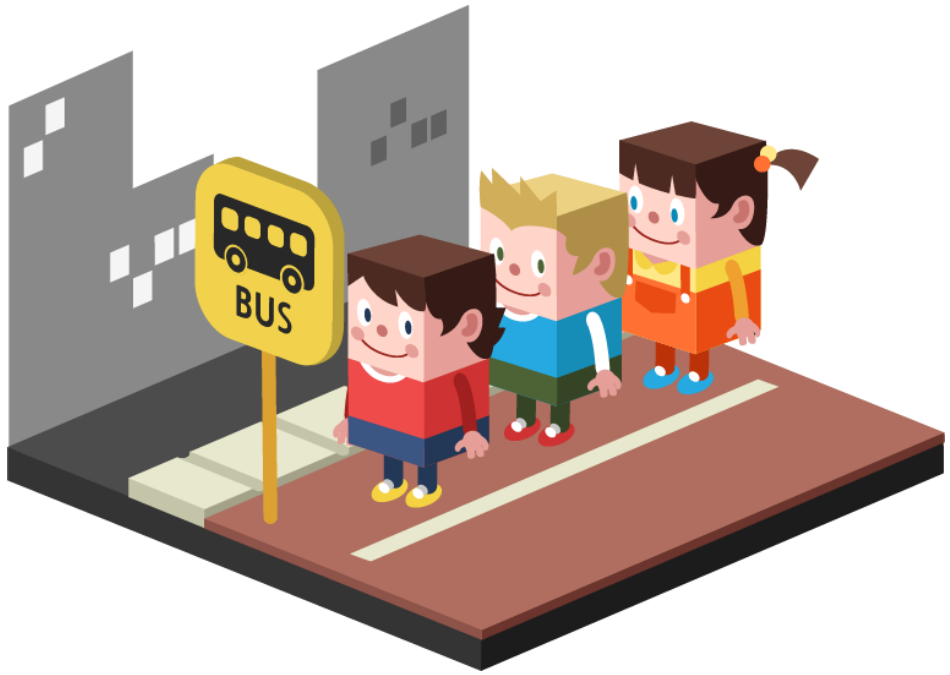
- 삭제 연산을 수행할 때 발생함
- 스택에 데이터가 존재하지 않으면 삭제가 일어나지 않는 현상

스택의 동작


```
push(S, 'A')
push(S, 'B')
if not empty(S) then pop(S)
push(S, 'C')
push(S, 'D')
```



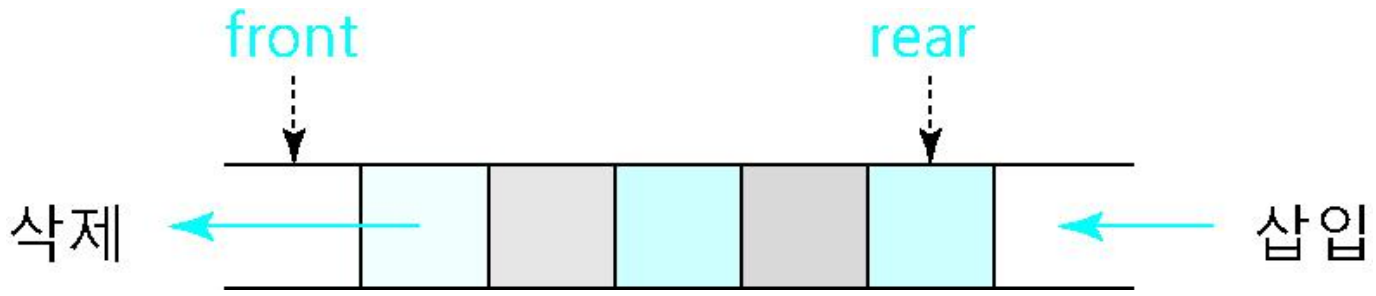
큐(Queue)



https://atoz-develop.tistory.com/entry/%EC%9E%90%EB%A3%8C%EA%B5%AC%EC%A1%B0-%ED%81%90-%EC%A0%95%EB%A6%AC-%EB%B0%8F-%EC%97%B0%EC%8A%B5%EB%AC%B8%EC%A0%9C

https://novoh2o.com.au/blogs/articles/how-do-water-coolers-work

큐(Queue)



큐(Queue)

개념

- 선형 리스트의 한쪽 끝에서는 데이터의 삭제만 이루어지고,
  다른 한쪽 끝에서는 데이터의 삽입만 이루어지는 자료구조
- 가장 먼저 입력된 데이터가 가장 먼저 제거되는
  선입선출(FIFO, First-In-First-out) 특징을 가짐

큐의 연산

오버플로 (overflow)

- 삽입 연산을 수행할 때 발생함
- 큐를 위해 할당된 저장 공간을 초과해서 더 이상 데이터를 삽입할 수 없는 현상

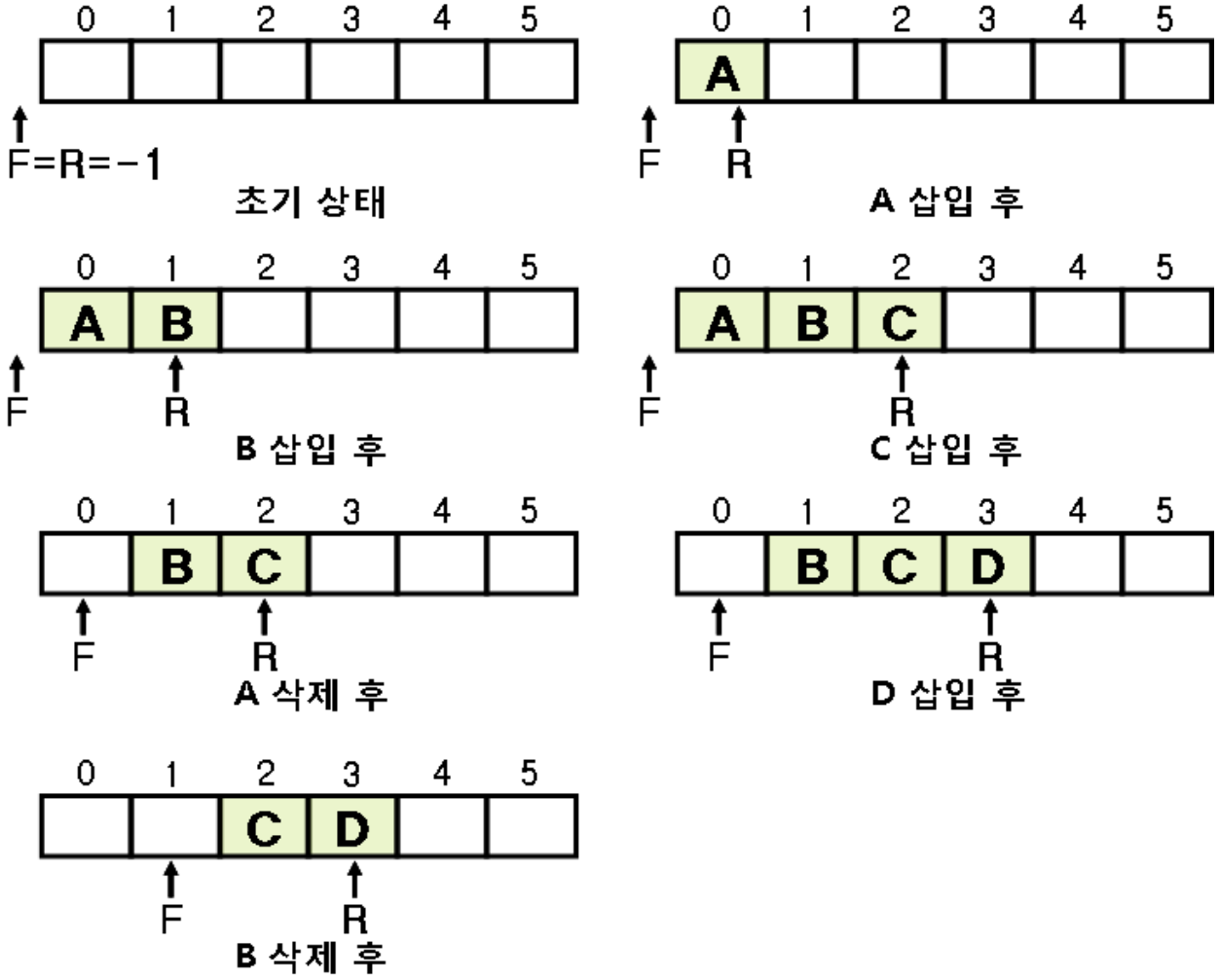
큐의 연산

언더플로 (underflow)

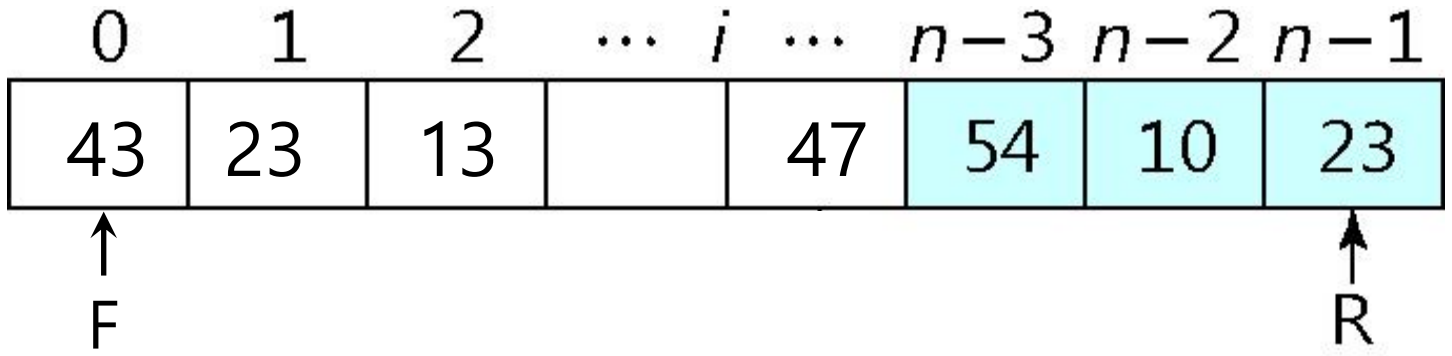
- 삭제 연산을 수행할 때 발생함
- 큐에 데이터가 존재하지 않으면 삭제가 일어나지 않는 현상

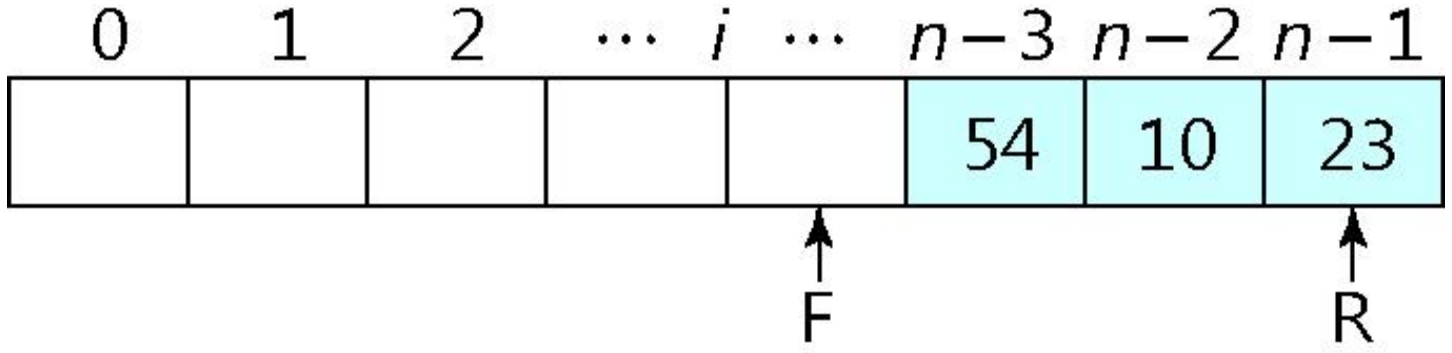
큐의 동작


```
enqueue(Q, 'A')
enqueue(Q, 'B')
enqueue(Q, 'C')
if not empty(Q) then dequeue(Q)
enqueue(Q, 'D')
if not empty(Q) then dequeue(Q)
```



연산 후의 큐의 상태 (만원 상태)





연산 후의 큐의 상태 (만원 상태)

만원 상태

- 데이터가 큐에 삽입됨에 따라 rear 변수 값이 증가하다가 n-1이 되면 더 이상 데이터가 삽입될 수 없는 상태가 됨
- 하지만, 이 경우가 반드시 큐에 n개의 항목이 가득 차 있다는 것을 의미하는 것은 아님
- 큐가 가득 채워진 상태를 결정하기 위한 다른 방법이 필요함

정리하기

1. 자료구조
   - 자료사이의 논리적관계를컴퓨터나프로그램이보다쉽게이해하고다룰수있도록구성한것
2. 배열
   - 같은자료형을갖는여러개의데이터를하나의변수로묶어놓은데이터의집합체이며, 각원소를구분하기위해인덱스와데이터값의쌍으로이루어짐
3. 연결리스트
   - 노드들을연결하여구성하는것으로,한노드는데이터필드와링크필드로구성됨
4. 스택
   - 리스트의한쪽끝에서만삽입과삭제가이루어지는후입선출(LIFO)구조
5. 큐
   - 리스트의한쪽끝에서는삽입,다른한쪽끝에서는삭제가이루어지는선입선출(FIFO)구조

04 강

다음시간 안내

자료구조(2)