



2017-2학기

7장-데이터 구성

박종혁 교수 (컴퓨터공학과)

jhpark1@seoultech.ac.kr

<http://www.parkjonghyuk.net>

➤ 7장 데이터 구성

- 1 이름
- 2 리스트(List)
- 3 그래프
- 4 계층

➤ 조별과제

- 다음 주 강의 시간 발표

- 적절하게 이름짓는 것의 중요성을 이해한다.
- 컴퓨터 시스템의 메모리 안에 어떻게 데이터가 구성하는 지를 이해한다.
- 리스트, 트리, 그래프가 메모리에 어떻게 저장되는 지를 이해한다.
- 리스트, 트리, 그래프가 가계도, 도로지도, 차트와 같은 우리에게 익숙한 개념들과 어떤 연관관계를 가지는 지를 이해한다.
- 인덱싱(indexing)이 메모리 안의 데이터를 구성하는데 어떻게 사용되는 지를 이해한다.
- 연결(linking)이 메모리 안의 데이터를 구성하는데 어떻게 사용되는 지를 이해한다

❖ **아이템이 이름을 적절하게 붙이는 것은 방대한 양의 아이템을 다룰 때
아이템의 유용성을 높인다.**

- 부적절한 아이템의 이름은 혼란을 유도한다.

❖ **이름 짓는 가이드 라인**

- 유일해야 한다.
- 둘 이상의 아이템이 같은 이름을 가지면 안된다.
- 하나의 아이템이 하나 이상의 이름을 가져서는 안된다.
- 이름은 서술적이어야 한다.

예) The Star Spangled Banner.mp3 (vs zqiy.mp 보다 좋다)

- 항목 (item)의 위치와 관련이 있어야 한다.

❖ 비 공식적으로 웹 주소 (web address) 또는 공식적으로 URL (Uniform Resource Locator)라고 불림

❖ 웹 주소의 가이드라인

- en.wikipedia.org/wiki/Coral_snake
 - 유일함
 - 이름을 하나만 가지고 있음
 - 설명적임 (Coral snake 에 대한 웹 페이지임을 알 수 있음)
 - 위치를 알 수 있음. Wikipedia 서버에서 wiki 라는 폴더 Coral_snake라는 파일을 나타냄.

- ❖ 많은 자료들이 리스트의 형태로 구성되어 있다.
- ❖ 리스트 (List) – 특정한 순서로 배치된 일련의 항목
- ❖ 예) 가장 값 비싼 그림 5 개의 리스트

1. 폴 세잔의 The Card Player
2. 잭슨 폴락의 No. 5, 1948
3. 월렘 드 쿠닝의 Woman III
4. 구스타프 크림트의
Portrait of Adele Bloch-Bauer I
5. 빈센트 반 고흐의
Portrait of Dr. Gachet



❖ 인덱싱 (indexing)

- 데이터의 각 아이템마다 유일한 번호를 붙이는 것
- 리스트의 각 아이템들은 인덱스 숫자에 의해 식별됨
- 대괄호 ([])를 사용하여 리스트의 아이템을 표현함
- Paintings[0], Paintings[1],...

❖ 컴퓨터안의 모든 데이터는 메모리 안의 특정위치에 저장됨

- 메모리 위치는 0부터 시작

❖ 리스트는 테이블처럼 표시되지만(왼쪽) 실제로는 하드 디스크 안에서 아래 그림과 같이 저장됨.

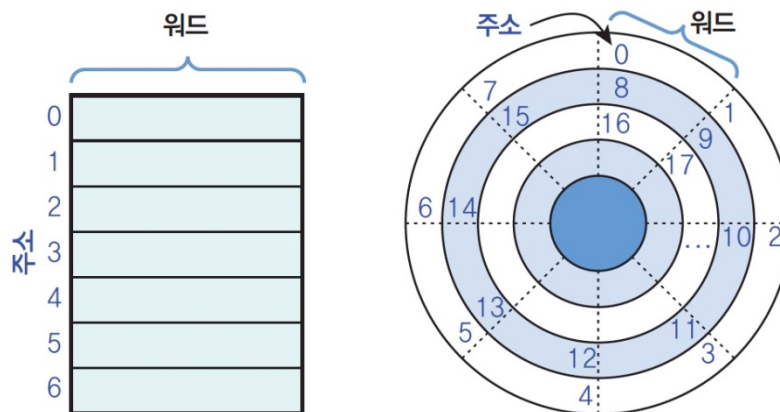


그림 7.1 메모리는 선형으로 구성된다. 메모리 데이터의 각 부분은 메모리 주소로 표시된 부분에 위치한다. 메모리 주소는 숫자 또는 인덱스로 표현된다. CD와 DVD 메모리 역시 선형으로 구성된다.

❖ 워드

- 컴퓨터에 저장되는 가장 작은 단위
- 8비트, 32비트, 64비트 등 컴퓨터마다 다르게 사용됨

❖ 트랙 (track)

- 디스크를 구성하는 동심의 원
- 디스크의 트랙을 따라가며 저장된 자료들을 읽음

❖ 섹터 (sector)

- 트랙을 구성하는 단위
- 각 섹터마다 한 워드 크기의 자료를 저장함

- ❖ 배열 - 메모리내의 리스트를 저장하기 위한 가장 간단한 저장 방법
- 프로그램에서는 왼쪽과 같이 저장되며, 하드 디스크에서는 오른쪽과 같이 저장된다.

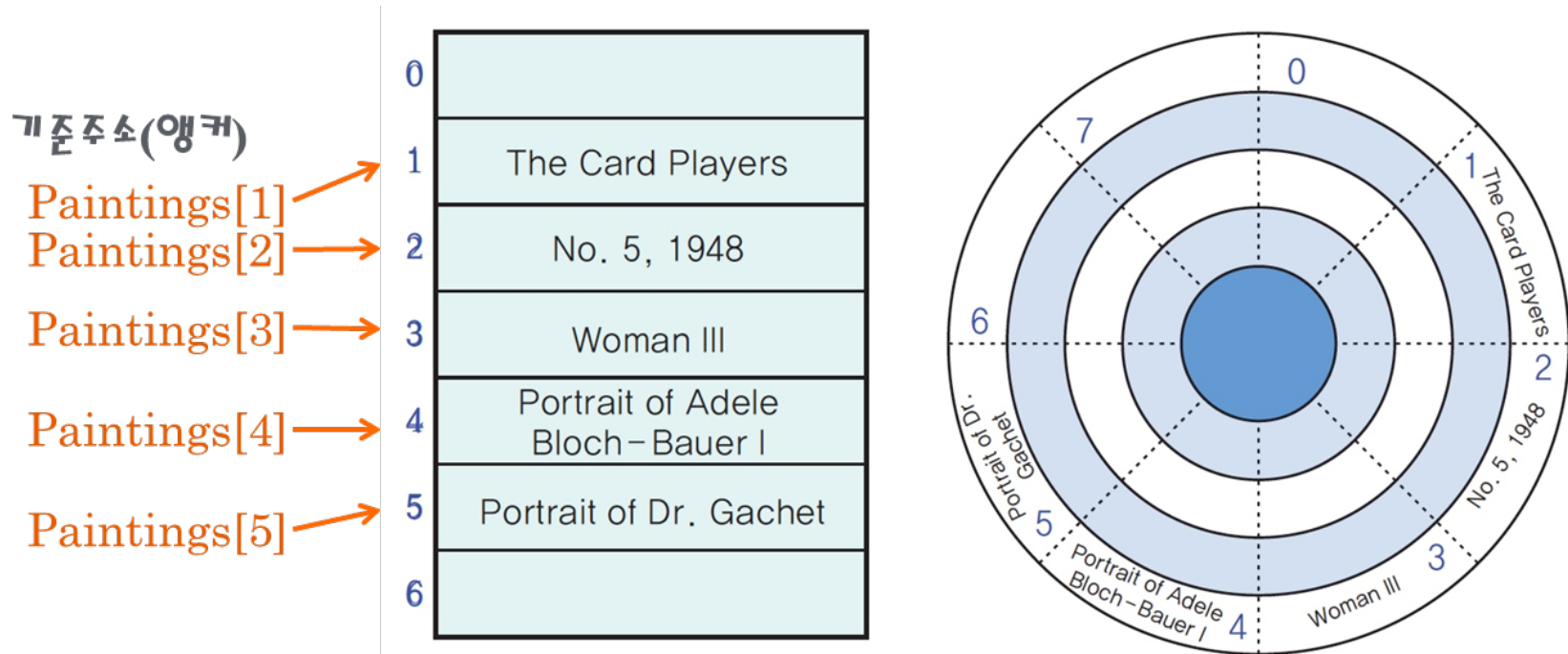


그림 7.2 다섯 그림의 리스트는 메모리 또는 DVD의 배열로서 저장된다.

❖ 배열의 위치는 기준 주소 또는 앵커라 불림

- 배열이 메모리에 저장되어 있는
메모리 상의 위치

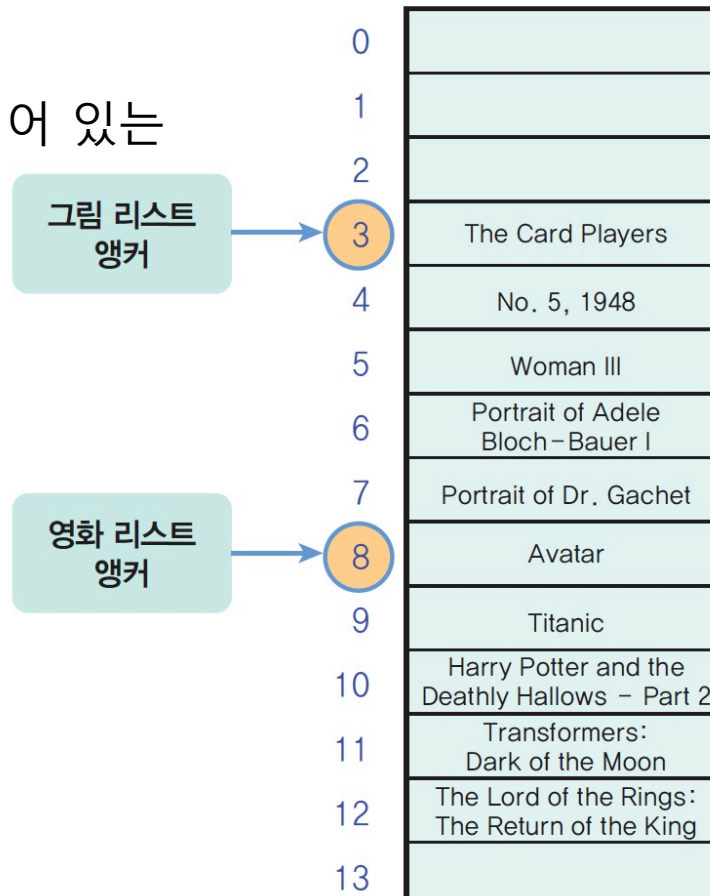


그림 7.3 메모리에서 두 배열이 배열로 저장되어 있다. 리스트 A가 메모리 주소 3부터 저장되어 있으며 리스트 B가 메모리 주소 8부터 저장되어 있다.

❖ 배열의 아이템 접근

- 기준주소 + 인덱스에 의해 아이템을 접근

❖ 배열의 장점

- 배열의 어떤 아이템도 자시의 리스트 안의 위치를 알고 있으면 쉽게 접근가능

❖ 배열의 단점

- 배열의 크기 고정
 - 리스트 내보의 삽입과 삭제를 위한 상당한 수의 연산을 필요로 함

❖ 배열의 아이템 접근

- 첫 아이템 - Paintings[1] 호출
- 배열의 3 번째 아이템 접근 - Paintings[3] 호출

❖ 컴퓨터가 리스트의 i 번째 아이템 주소 계산 방법

- $\text{Paintings}[i] = \text{Paintings의 앵커 주소} + (i - 1)$

❖ 0 인덱싱

- i 에 -1 을 해줘야 하므로 연산이 많아짐. 따라서 배열의 첫 아이템은 0으로 시작하게 함.
- $\text{Paintings}[0]$ 을 호출하여 배열의 첫 아이템을 접근
 - $\text{Paintings}[i]\text{의 주소} = \text{Paintings의 앵커 주소} + i$

7.2 리스트 (배열 자료 삭제)

- ❖ 삭제하려는 아이템을 제거 후, 그 아이템 다음에 위치한 모든 아이템을 하나씩 옮김
- ❖ 삭제에 시간이 많이 걸림

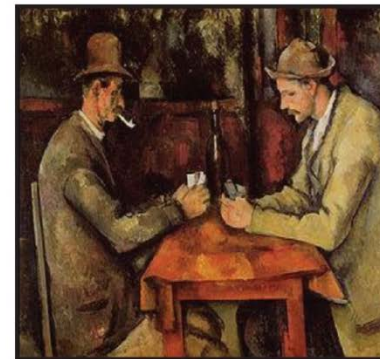
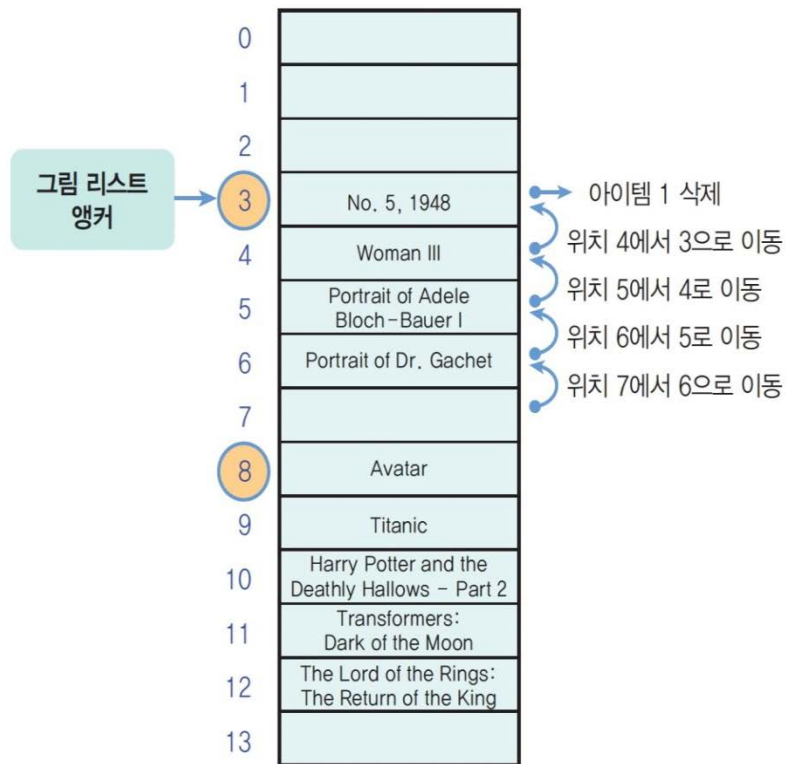


그림 7.5 그림 리스트에서 첫 아이템을 삭제하기. 이를 위해서는 메모리의 4개의 아이템 이동이 필요하며 리스트 앞의 아이템에서부터 리스트 뒤의 아이템의 순서로 이동을 진행한다.

7.2 리스트 (배열 자료 삽입)

- ❖ 삽입하려는 위치 및 그 아래에 있는 모든 아이템을 옮긴 후 삽입
- ❖ 삽입에 시간이 많이 걸림

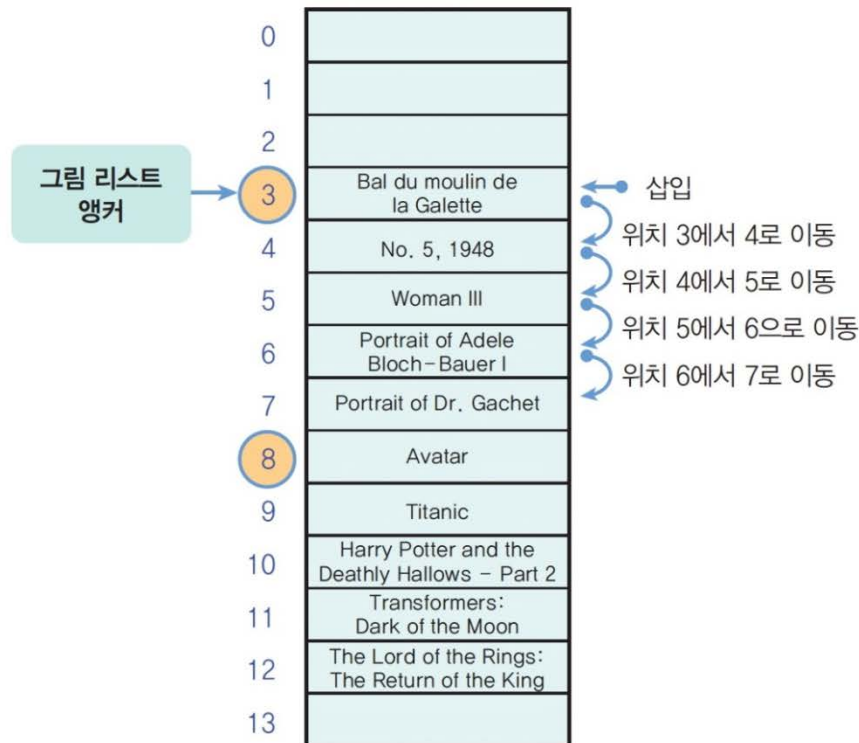


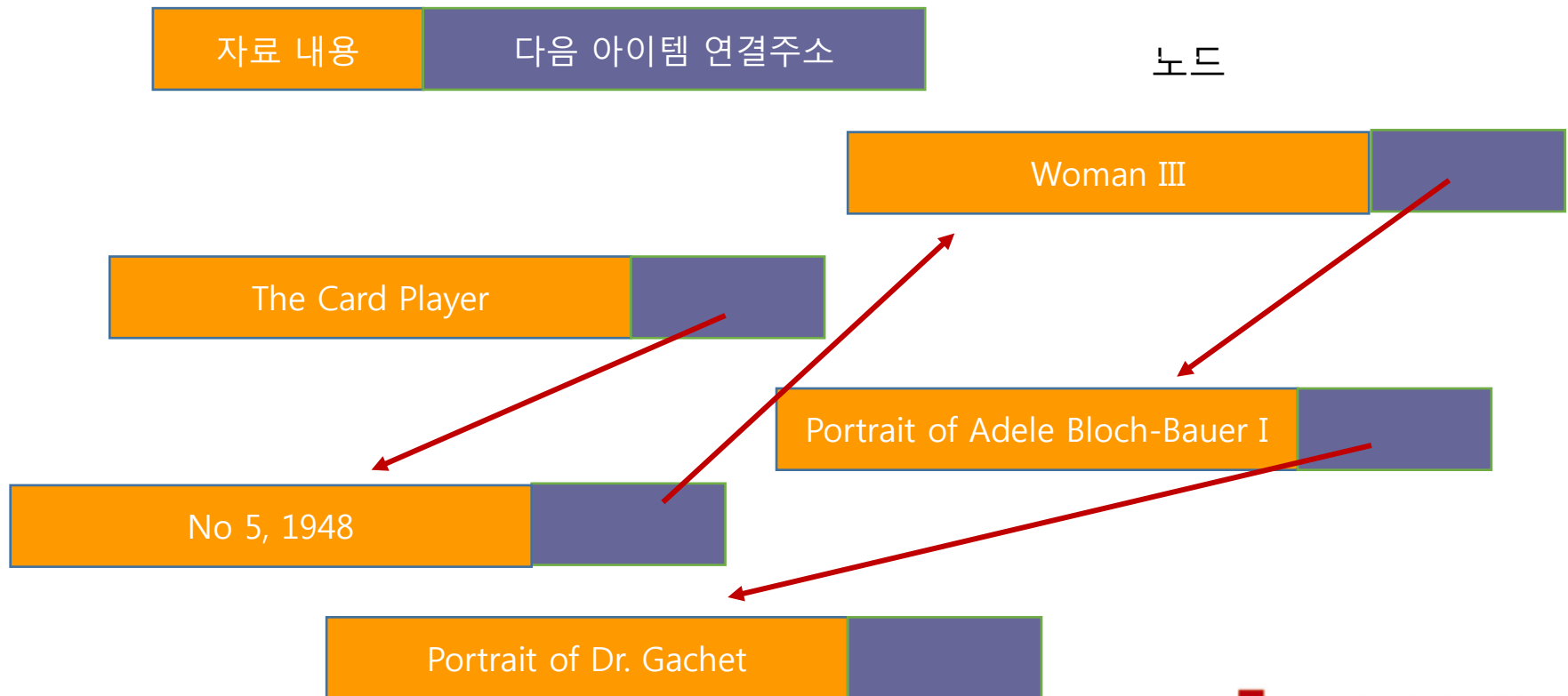
그림 7.6 리스트의 앞에 아이템을 삽입하기. 이 경우 리스트 내의 모든 다른 아이템들이 새로운 아이템에 자리를 내주기 위해서 이동해야 한다.

- ❖ 메모리 안의 데이터를 연결하는 것 - 메모리 내의 리스트를 저장하는 가장 일반적 방법
- ❖ 연결 리스트 (linked list): 메모리 안에 흩어져 있는 데이터를 사슬처럼 연결하여 리스트를 구성하는 자료 구조
- ❖ Geocache 시합 (geocaching의 변화된 게임)
 - 데이터를 연결하는 기본적인 아이디어 geocaching 게임과 유사
 - 보물 + 다음 GPS 좌표



❖ 연결 리스트 (linked list): 메모리 안에 흩어져 있는 데이터를 사슬처럼 연결하여 리스트를 구성하는 자료 구조

❖ Paintings 연결 리스트



7.2 리스트 (연결 리스트)

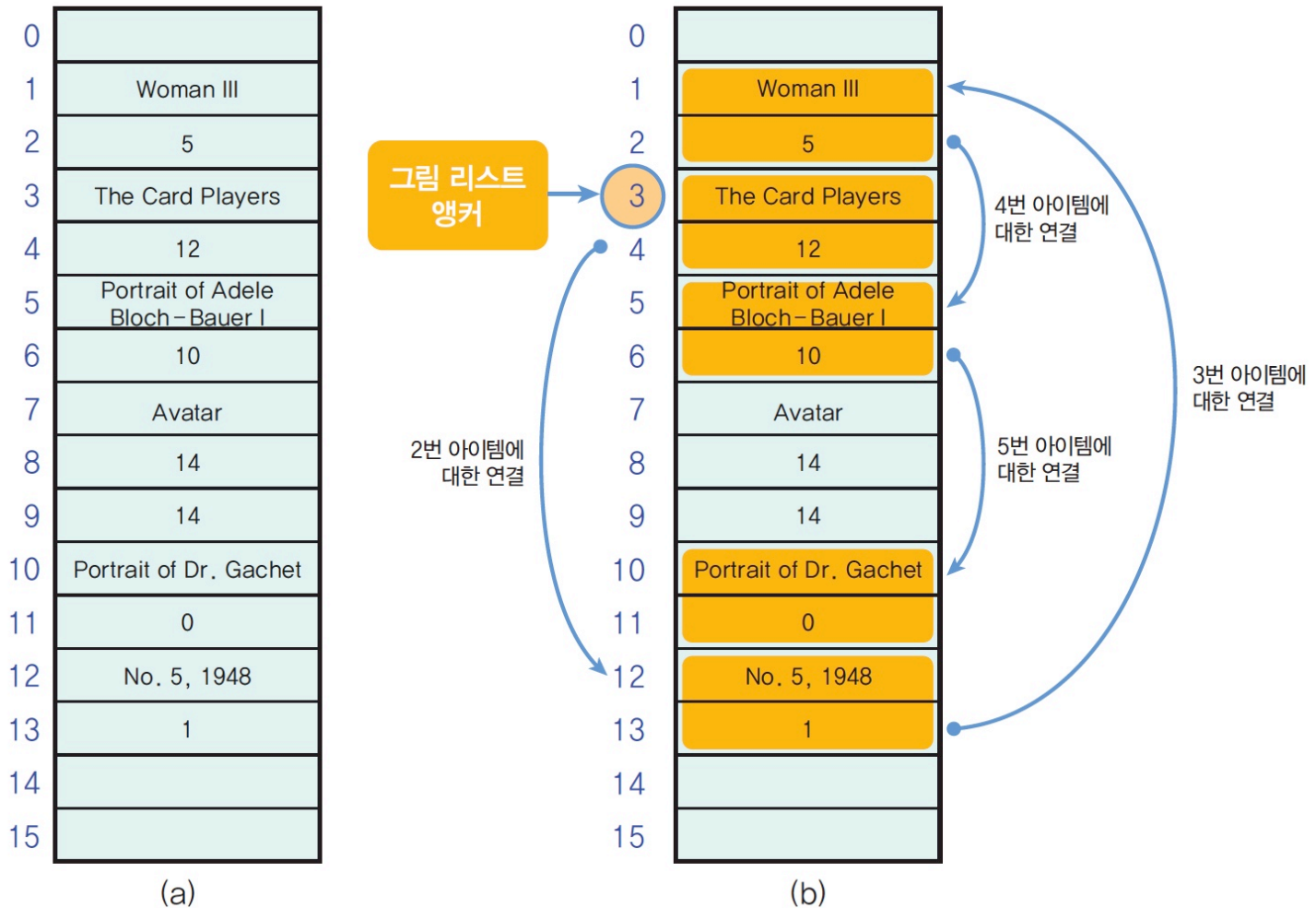


그림 7.8 다섯 개의 그림 리스트를 아이템을 연결 방법을 사용하여 메모리에 저장하기.

(a) 메모리 상태 (b) 메모리 상태에 대한 해석

- ❖ 앵커와 인덱스 만으로 직접 접근이 안됨
- ❖ 앵커에서 출발 인덱스 숫자 만큼 다음 아이템 연결 주소로 이동해야 함.
- ❖ 연결 리스트 자료 삽입

1. 삽입할 아이템에 대한 메모리 공간 확보

- 아이템 데이터 + 아이템 다음 연결 주소가 쓰여지는 공간

2. 새 아이템의 다음 연결 주소는 삽입할 위치 앞의 아이템이 가지고 있는 연결 주소로 해준다.

3. 삽입할 위치 앞의 아이템의 연결 주소를 새 아이템의 메모리 주소로 바꾸어 준다.

7.2 리스트 (연결 리스트의 자료 삽입)

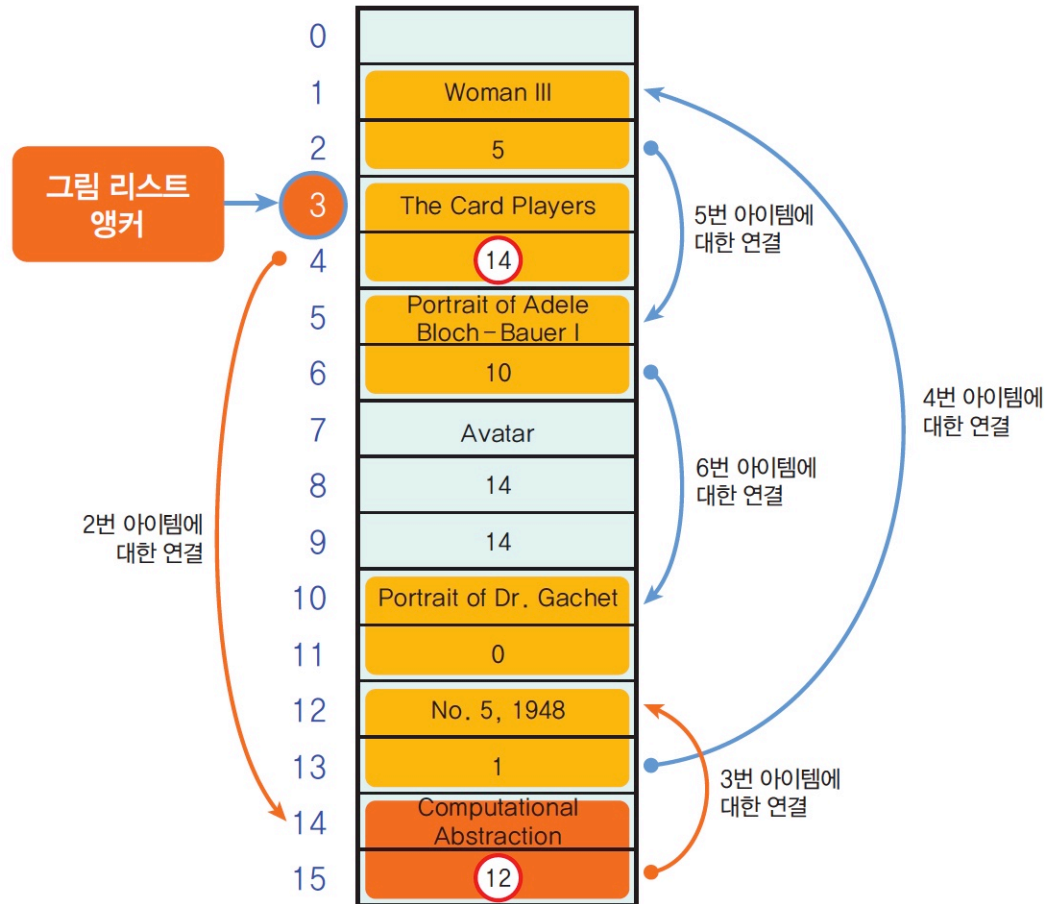


그림 7.9 리스트에 새 요소를 삽입하기 위해서는 새로운 노드(셀 14, 15)를 만들어야 하며 새로운 노드를 가리키기 위해 기존의 연결 상태를 변경시켜야 한다.

❖ 연결 리스트 자료 삭제

1. 삭제할 아이템에 대한 메모리 주소 확보
2. 삭제할 아이템 바로 앞의 아이템 메모리 주소 확보
3. 앞 아이템의 다음 연결 주소를 삭제할 아이템의 연결 메모리 주소로 해준다.

- ❖ 실생활의 많은 객체들과 개념들은 그래프로 모델링 됨
- ❖ 정점들과 정점을 잇는 간선들의 집합
 - 정점 (node): 항목, 간선(edge): 두 개의 정점을 이어주는 관계
- ❖ 방향성 그래프 (directed graph)
 - 모든 간선들이 방향성을 가진 그래프
 - 방향성을 가진 간선을 아크 (arc)
 - $G = (V, E)$
 - // g 는 정점들의 집합 V 와 아크들의 집합 E 로 구성됨

- ❖ $V = \{A, B, C, D, E\}$
- ❖ $E = \{(A,E), (A,B), (B,A), (B,D), (C,E), (D,C), (E,B), (E,C), (E,D)\}$
- ❖ $G = \{\{A, B, C, D, E\}, \{(A,E), (A,B), (B,A), (B,D), (C,E), (D,C), (E,B), (E,C), (E,D)\}\}$

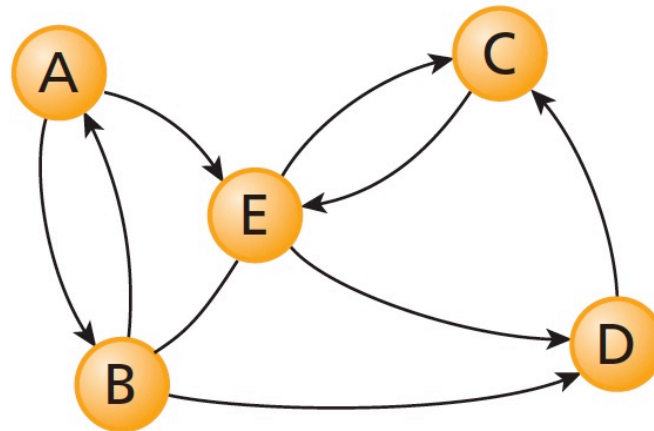


그림 7.10 그래프 G가 도형으로 그려진다. G는 다섯 개의 정점과 아홉 개의 아크를 가지고 있다.

7.3 그래프 (그래프 예시)

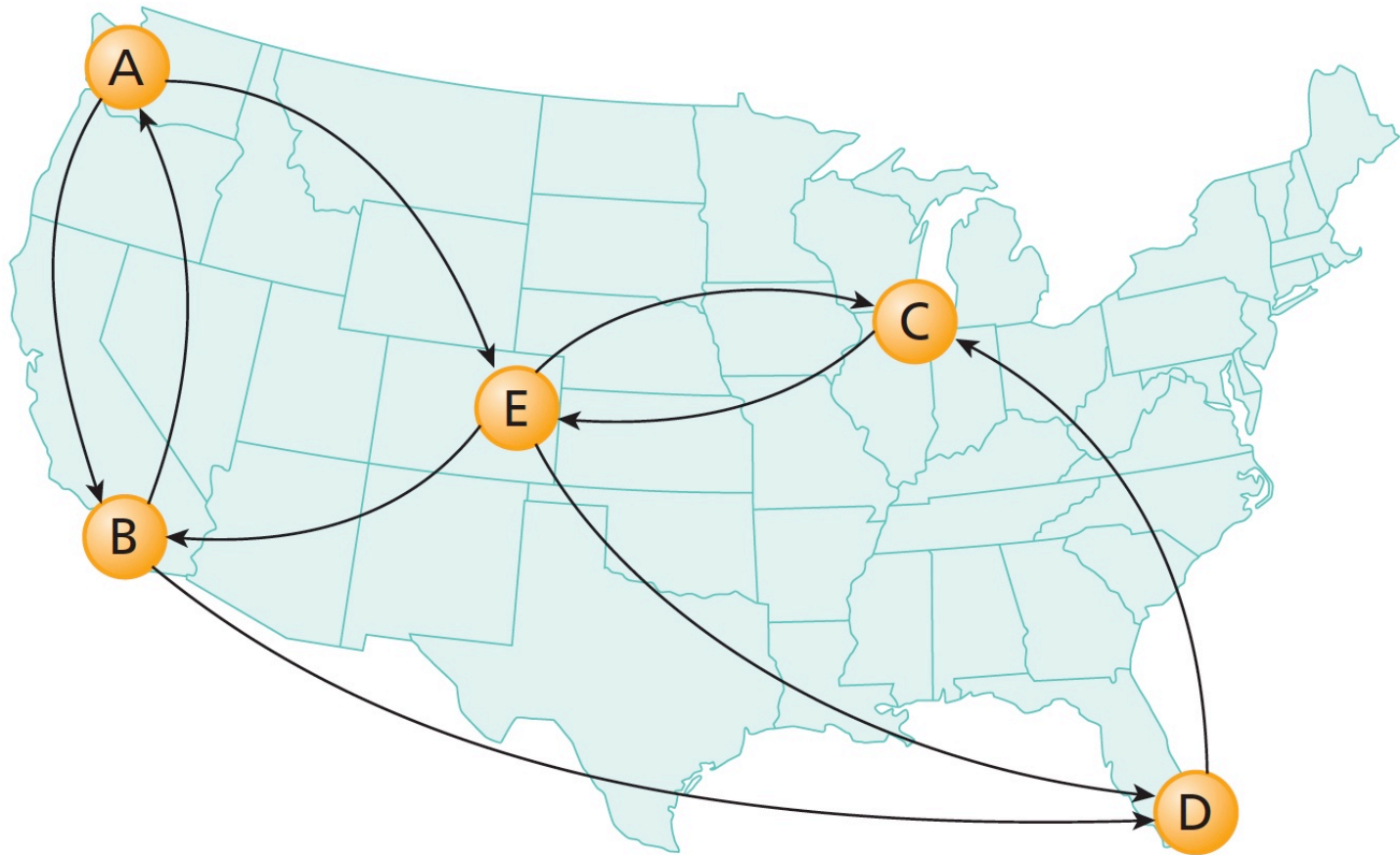


그림 7.11 미국내 다섯 도시 간의 항공편을 모델링한 그래프.

A: 시애틀, B: LA, E: 덴버, C: 시카고, D: 마이애미

7.3 그래프 (그래프 예시)

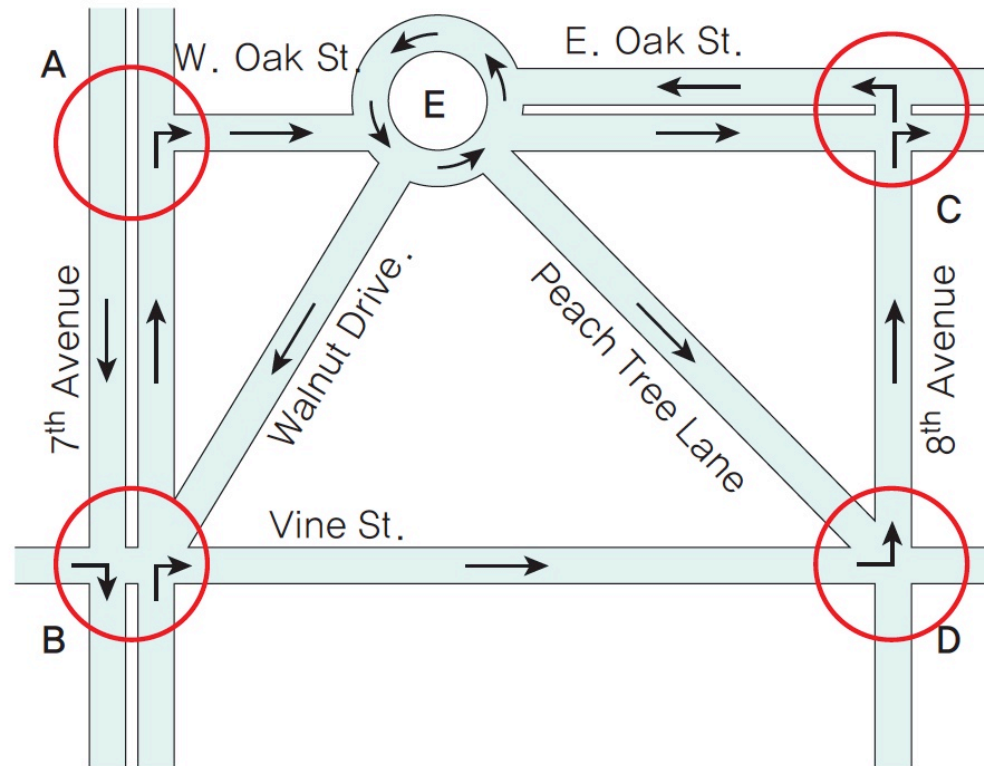


그림 7.12 도로 지도를 모델링하는 그래프. 각 정점은 교차로를 나타내고 각 아크는 교차로 사이의 도로를 나타낸다.

❖ 실제 세계의 대부분의 상황들은 객체와 객체들간의 관계로 추상화할 수 있다.

- 소셜 네트워킹
 - 객체 - 사람, 관계 - 친구관계
- 지도
 - 객체 - 지점, 관계 - 도로
- 수강 신청
 - 객체 - 교과목, 학생
 - 관계 - 학생-교과목 수강신청 상황

❖ 실제 문제를 해결하기 위한 추상화 과정에서 그래프는 유용하게 사용된다.

- 객체 - 정점
- 관계 - 간선

7.3 그래프 (그래프의 용어와 특성)

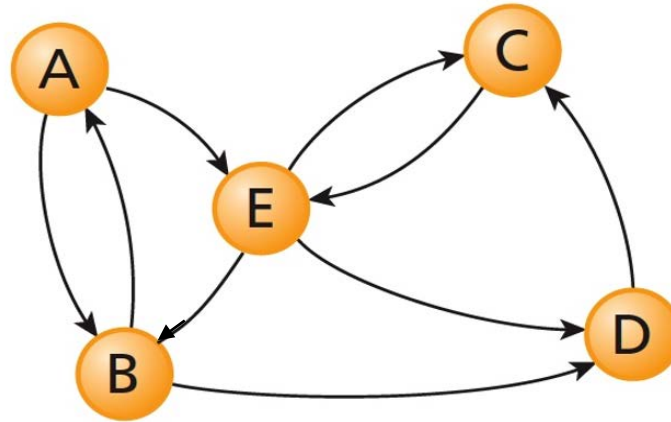


그림 7.13 그래프 G

- ❖ **인접성 (Adjacency)** – 두 정점 사이에 간선이 있을 때 두 정점은 인접해 있다고 말한다.
 - 예) 아크(D, C)가 G에 있다면 정점 D는 C에 인접한다
아크(C, D)는 G에 없기 때문에 정점 C는 D에 인접해 있지 않다
- ❖ **루프(Loop)** – 간선의 시작 정점과 도착 정점이 일치할 때 간선은 루프를 구성한다.
- ❖ **입력차수(In-degree)** – 정점 v 를 도착 정점으로 하는 간선의 수를 v 의 입력차수라고 한다.
 - 정점 E의 입력차수 2: E를 두 번째 정점으로 하는 아크 (A, E), (C, E)
- ❖ **출력차수(Out-degree)** – 정점 v 를 시작 정점으로 하는 간선의 수를 v 의 출력차수라고 한다.
 - 정점 E의 출력차수 3: E가 처음 정점인 아크 (E, B), (E, C), (E, D)

7.3 그래프 (그래프의 용어와 특성)

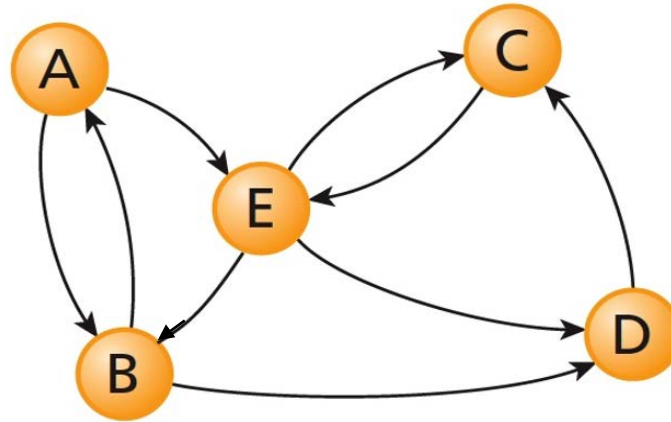


그림 7.13 그래프 G

- ❖ **위수(Order)** – 그래프 정점들의 갯수 (예제 – 위수 : 5)
- ❖ **크기(Size)**– 그래프 아크들의 개수 (예제 – 크기 : 9)
- ❖ **경로(Path)** – 그래프 내에서 아크로 이어지는 정점들의 집합
 - 예) [A, E, C]는 경로, [A, B, E] – 경로 아님
- ❖ **경로 길이(Path length)** – 경로에 있는 아크의 숫자
 - 예) [A, E, C] 경로 길이: 2
- ❖ **사이클(Cycle)** – 길이가 0보다 크며 처음과 마지막 정점이 일치하는 경로
 - 비순환적 그래프: 사이클이 없는 그래프
 - 예) [A, B, A]는 사이클

그래프 G는 비순환적이 아님

7.3 그래프 (저장)

- ❖ 메모리에 앵커, 출력차수, 인접한 정점들의 메모리 주소를 표시함
- 메모리 14에 저장된 정점 'E' 참고

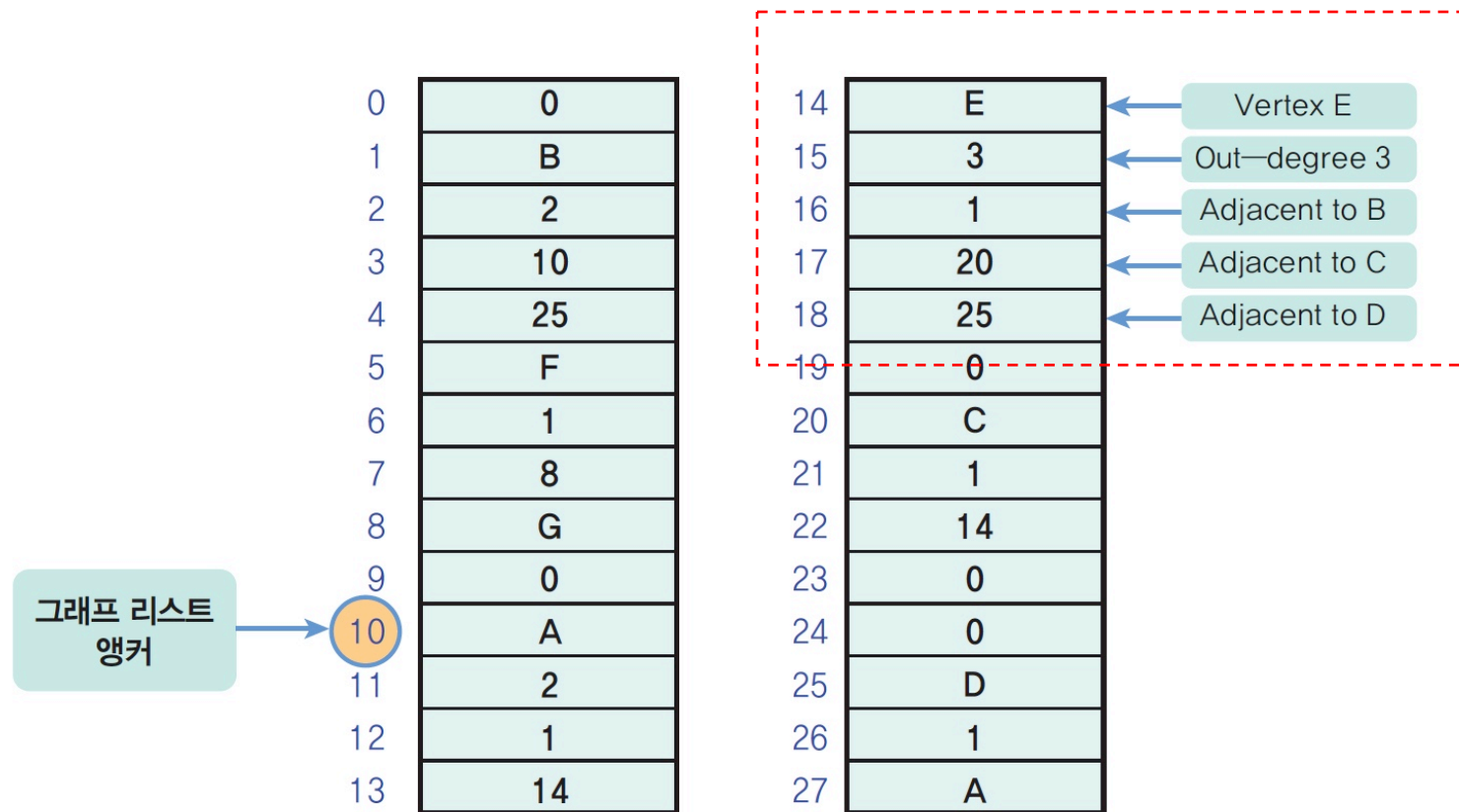


그림 7.14 그래프 G는 선형 메모리에 저장된다.

❖ 요소들이 레벨 별로 배치되도록 하는 방법

- 각 구성원은 바로 밑에 다수의 구성원을 가질 수 있으나 위에는 하나의 구성원 만을 가지도록 제한됨.

❖ 많은 컴퓨터 상의 개념들은 구성원들을 계층화함에 의해 생성됨

❖ 여러 실제 자료들이 계층화되어 컴퓨터에 저장됨

❖ 예 - 조직차트, 가계도, 생물학, 언어학, 트리

❖ 조직차트 - 조직의 구조를 나타내는 다이어그램

- 조직원의 관계 및 보고체계를 보여줌

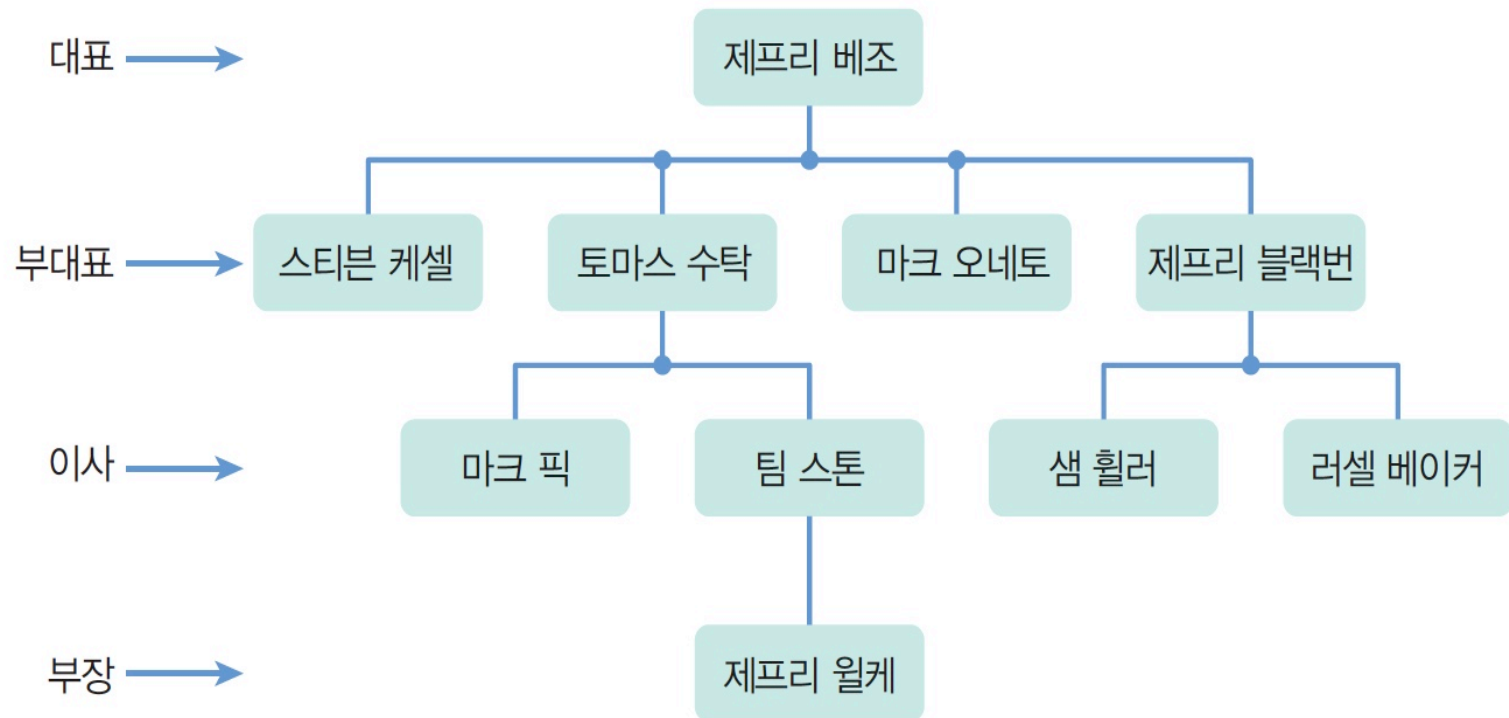


그림 7.15 Amazon.com의 간략화된 조직 차트는 회사 조직 내의 4 단계 직급 체계를 보여준다.

❖ 가계도 - 조상과 후손들 사이의 유전적인 관계를 나타내는 그림

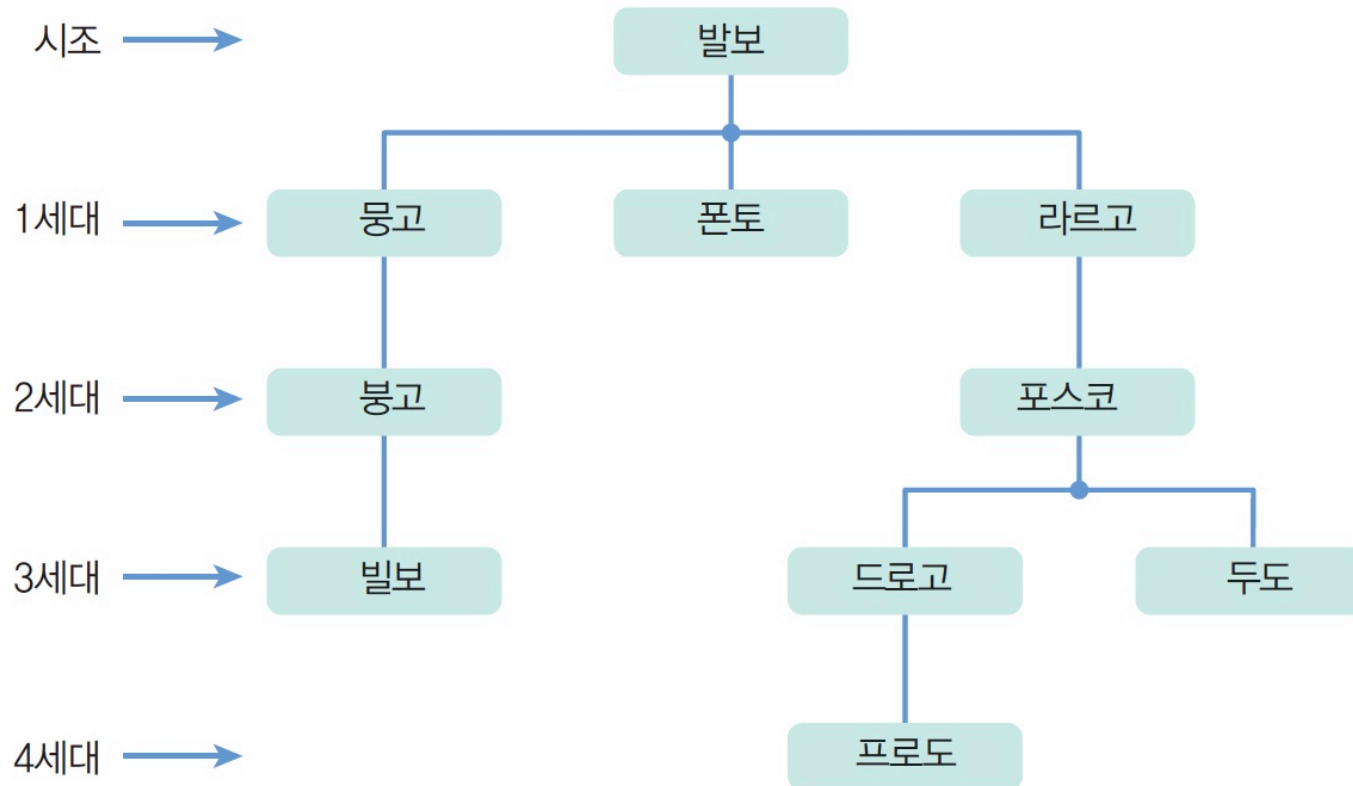


그림 7.16 발보 배긴스와 프로도 배긴스의 조상을 보여주는 간략화된 가계도

❖ 분류법 (taxonomy)

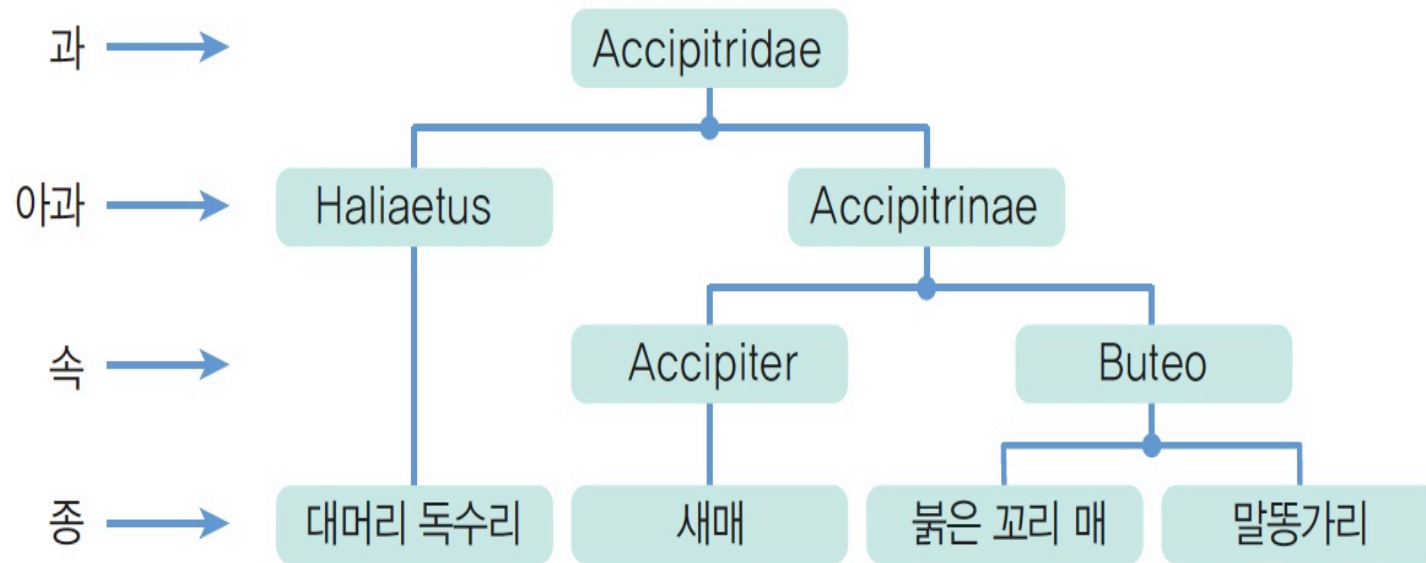


그림 7.17 네 종류의 새에 대한 분류법

❖ 구문해석 트리

- 언어 문법을 모델링하고 언어 요소들의 의미를 명확하게 문법을 표현해주는 도구

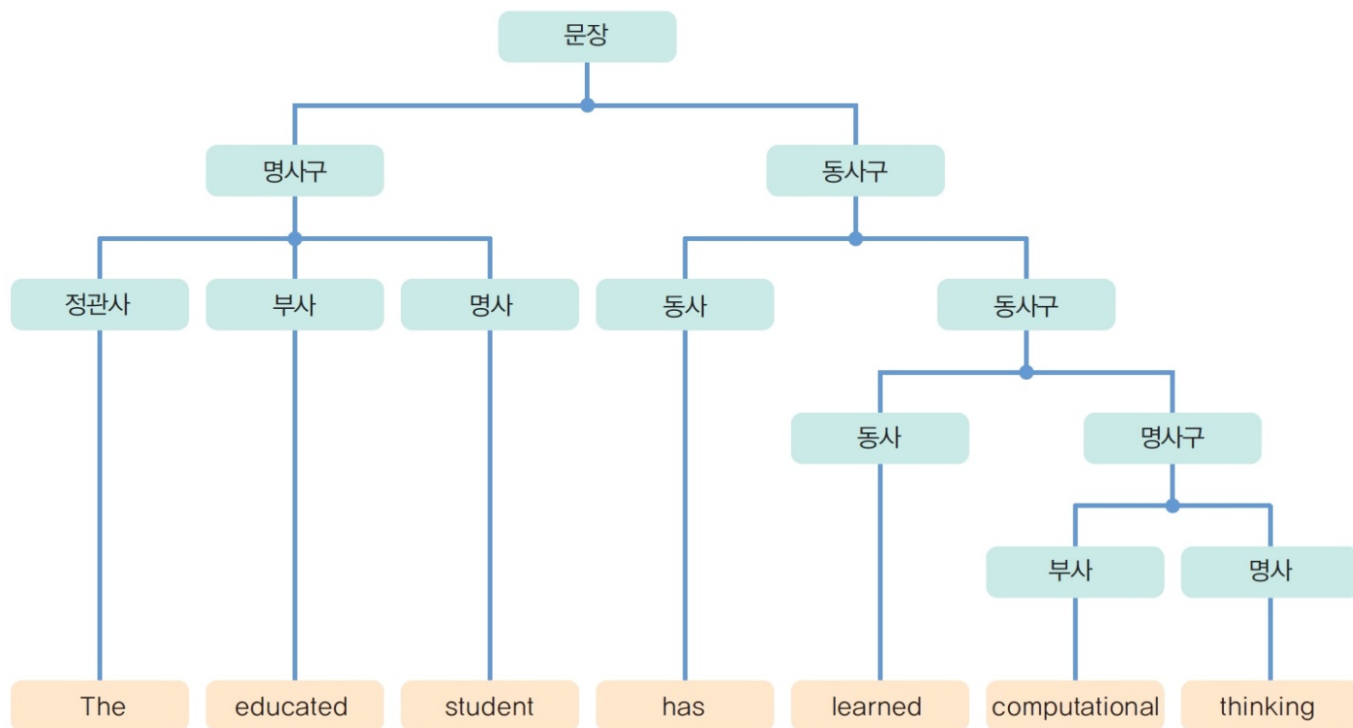


그림 7.18 구문해석 트리에서 보여진 영문 문장의 구조. 이 문장은 하나의 명사구와 하나의 동사구로 이루어져 있다. 이 구문 각각은 다른 문법적인 요소들로 구성된다.

❖ 계층 데이터를 모델링하기 위한 그래프의 한 형태

❖ 트리의 특성

- 그래프의 입력 차수가 0인 정점이 정확히 하나 (루트라고 불림)
- 그래프의 루트가 아닌 모든 정점은 입력 차수가 1
- 루트에서 다른 정점들 모두에게 도달하는 경로가 있음

❖ 말단 정점

- 트리에서의 출력 차수 0의 정점

❖ 트리의 루트

- 입력 차수가 0인 정점
- 오른쪽 그림의 정점 'R'

❖ 말단 정점

- 출력 차수가 0인 정점
- 오른쪽 그림의 정점 'T', 'E', 'C', 'A'

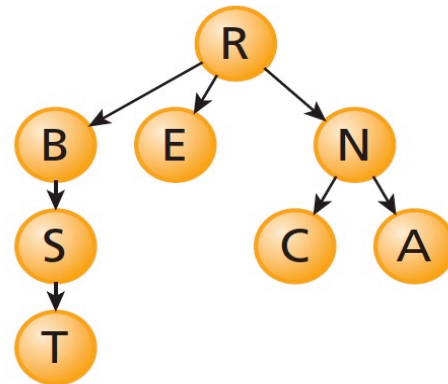


그림 7.19 트리는 관례적으로 루트(root)가 위로 리프(leaf)가 아래에 위치한 위아래가 뒤집힌 채로 그려져 왔다.

- 발표방식: ppt 자료 활용 2개 조 각각 10분 발표 (질의응답 포함)
- 문제 1
 - 우리가 생활하면서 모든 것은 자신의 생활 패턴에 대해 정리되어 있으며, 자신만이 식별가능 하다. 예를 들어 난장판이 되어있는 방을 정리 하려할 때 방 주인이 ' 이게 난장판인것 같이 보여도 나에겐 나만의 흐름으로 정리되어 있는거야 ' 이러한 말을 하는 것을 볼 수 있다. 여기서 방주인은 자신의 흐름 즉, 이름, 모양, 색 등을 기준으로 정리할 것이다. 이와 같이 정리된 흐름이라는 것을 하나의 데이터라고 하며, 이러한 데이터를 나열하는데 사용하는 방법으로 리스트, 트리, 그래프 등 다양한 방법이 있다. 위에 제시한 데이터 나열법에 대해 실생활에서 사용되는 예시를 하나 이상씩 조사하고 추후 적용되어질 수 있는 분야에 대해 생각해 보아라.
- 문제 2
 - 보통 우리가 사용하는 컴퓨터는 셀 수 없을 정도의 데이터를 처리한다. 이러한 데이터를 구별하기 위해 가장 많이 사용되고 있는 방식이 트리 구조를 이용하는 방식이다. 트리 구조에 대한 데이터 처리방식의 예를 들고, 트리 구조의 장단점이 무엇인지 설명하여라.