제 8장 Graph Theory

8.1 Introduction

8.2 Paths and Cycles

8.3 Hamiltonian Cycles and the Traveling Salesperson

8.4 A Shortest-Path Algorithm

8.5 Representations of Graphs

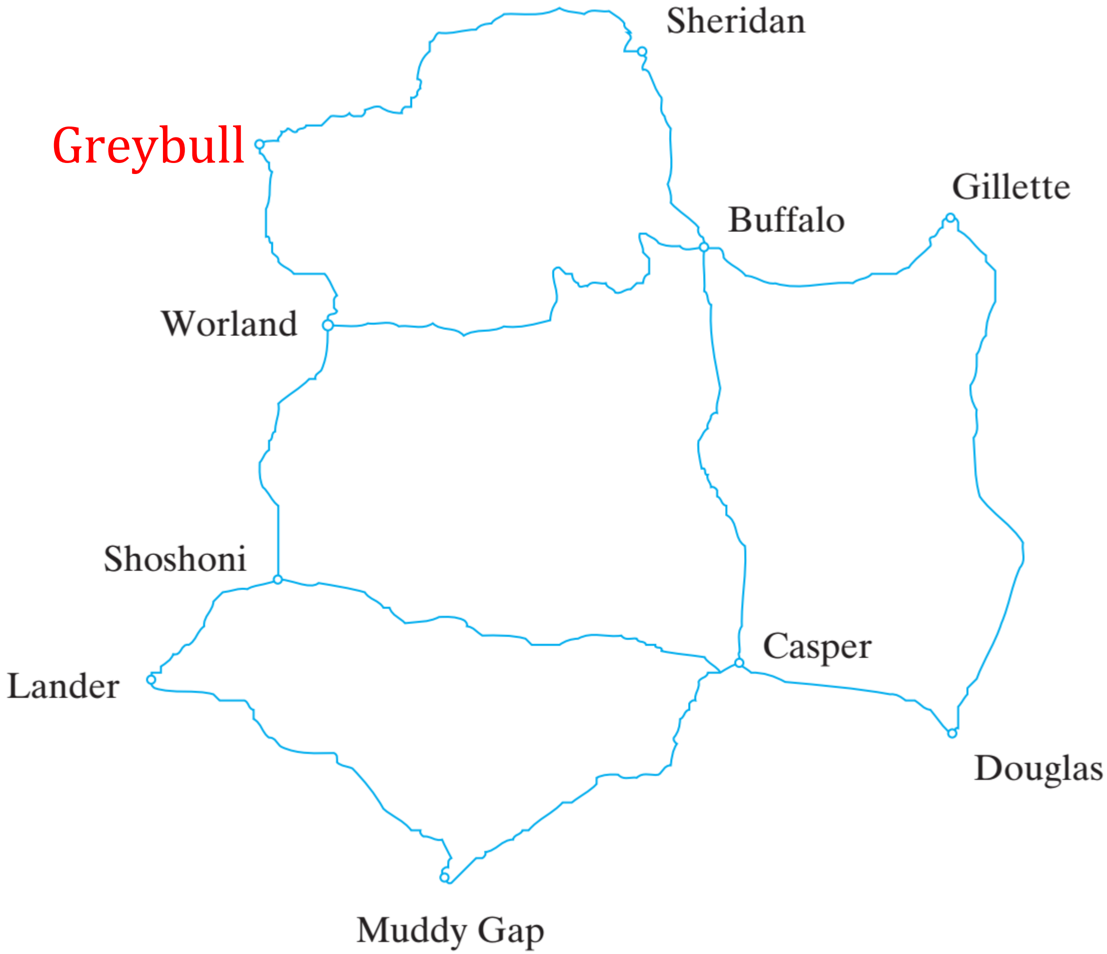
8.6 Isomorphisms of Graphs

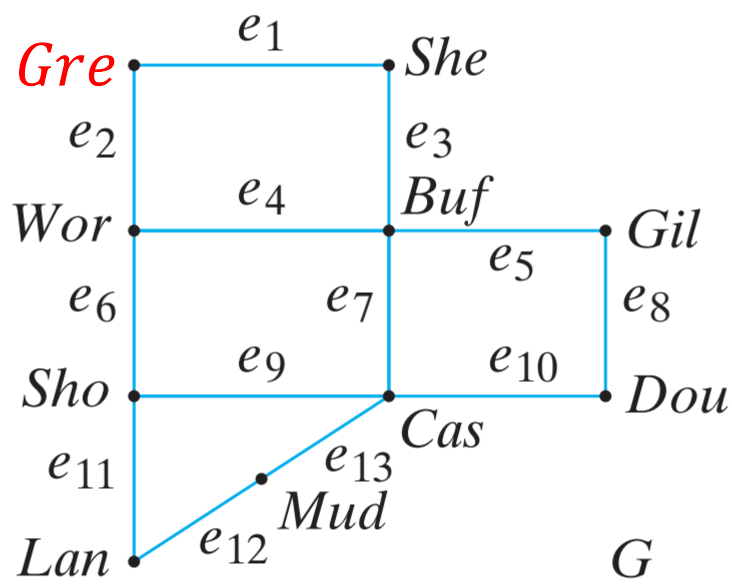
8.7 Planar Graphs

~~8.8 Instant Insanity~~

8.1 Introduction

- 도로 순찰차가 그레이불에서 출발하여 모든 도로를 정확히 한 번씩 지나 다시 그레이불로 되돌아오는 것이 가능할까?

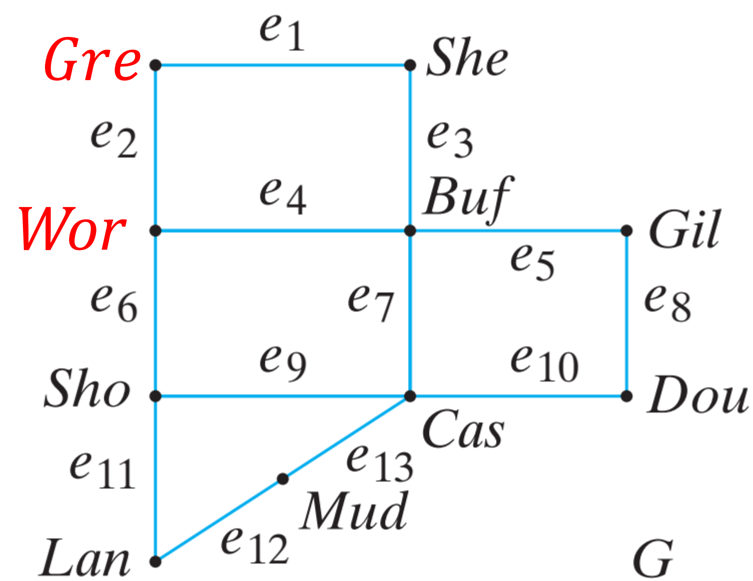




와이오밍 주의 고속도로

8.1 Introduction

- 지도를 **그래프**로 모델링. 정점Vertex: 도시, 간선edge: 도로
- 정점 v_0에서 출발하여 정점 v_1으로 가는 간선을 따라가고, 또 정점 v_2로 가는 또 다른 간선을 따라가고, 이런 식으로 하여 결국 정점 v_n에 도착한다. 이 때 v_0에서 v_n으로의 완전한 방문(순회)을 **경로**path라고 한다.
- 그림에서 정점 Gre에서 출발하여 모든 간선을 한 번씩만 지나서 정점 Gre로 되돌아오는 경로는 없다. 있다면,...
- 어떤 간선으로 Wor에 도착하면 다른 간선을 따라 반드시 떠나야 한다. 또한 Wor에 연결된 모든 간선은 반드시 사용되어야 한다. 짝수개의 간선이 Wor에 연결되어야 함을 뜻한다. 3 개의 간선이 Wor에 연결되어 있으므로 모순임을 알 수 있다.



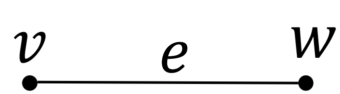
8.1 Introduction

- 정의 8.1.1 그래프(또는 **무방향 그래프** *undirected graph*) G는 정점의 집합 V와 간선의 집합 E로 구성되어 있고, 각각의 간선 $e \in E$는 순서가 없는 정점의 쌍으로 나타낸다.

  방향 그래프 *directed graph or digraph* G는 정점의 집합 V와 간선의 집합 E로 이루어져 있고, 각각의 간선 $e \in E$는 순서가 있는 정점의 쌍이다.

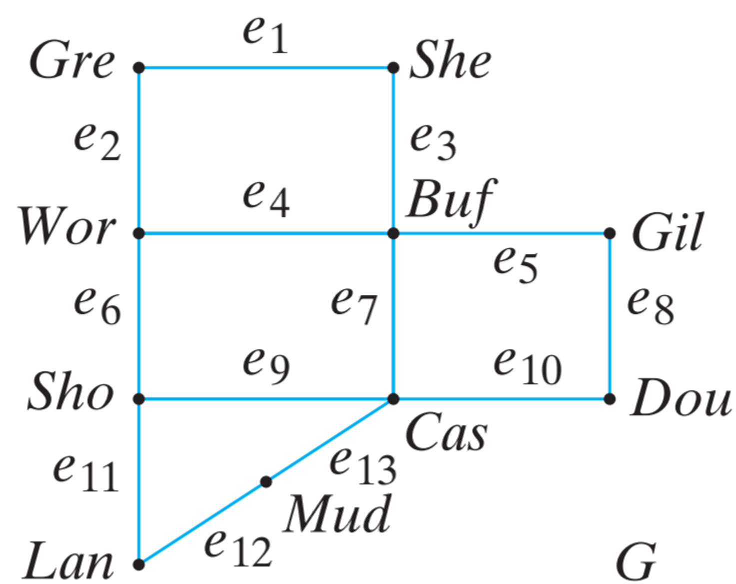
  그래프에서 정점의 쌍 v와 w를 잇는 간선 e는 v와 w에 결합된다 *incident on* 라고 한다. 그리고 정점 v와 w는 간선 e에 결합된다고 하며 인접 *adjacent* 정점이라 한다.

  G가 정점 V와 간선 E를 가지는 그래프이면 $G = (V, E)$라 표기한다



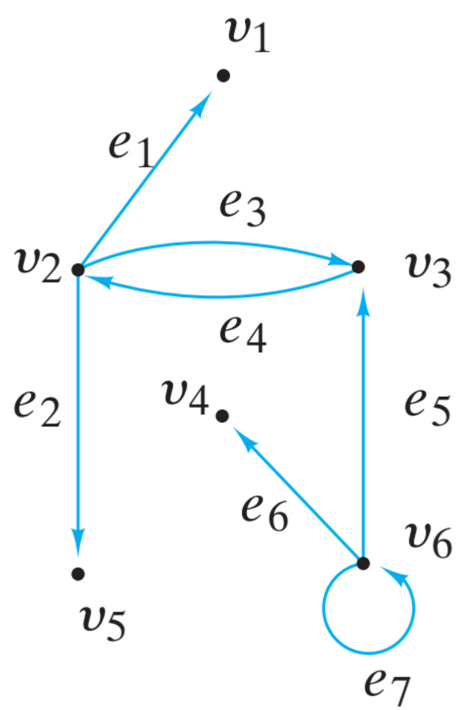
8.1 Introduction

- 예제 8.1.2 (무방향) 그래프 G는 정점의 집합
  $$V = \{Gre, She, Wor, Buf, Gil, Sho, Cas, Dou, Lan, Mud\}$$
  간선의 집합
  $$E = \{e_1, e_2, \dots, e_{13}\}$$
- 간선 e_1은 정점의 무순서쌍 $\{Gre, She\}$에 해당하고, (Gre, She) 또는 (She, Gre)라 표기
- 간선 e_{10}은 정점의 무순서쌍 $\{Cas, Dou\}$에 해당된다 (Cas, Dou) 또는 (Dou, Cas)라 표기
- 간선 e_4는 정점 Wor와 Buf에 결합되어 있고, 정점 Buf와 Wor는인접한다.



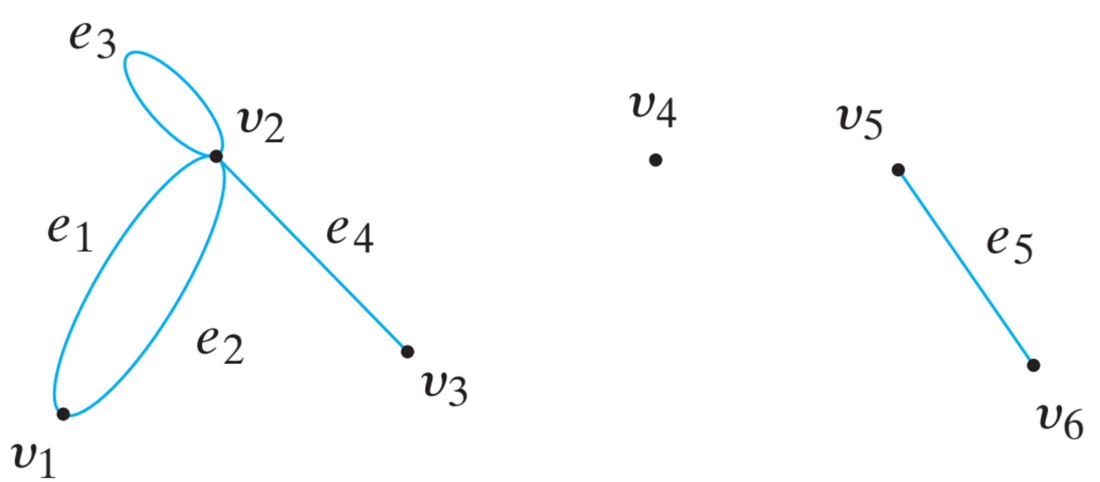
8.1 Introduction

- 예제 8.1.3 방향 간선은 화살표로 나타낸다.
- 간선 e_1은 정점 순서쌍 (v_2, v_1)으로 나타내고, 간선 e_7은 정점의 순서쌍 (v_6, v_6)를 나타낸다.



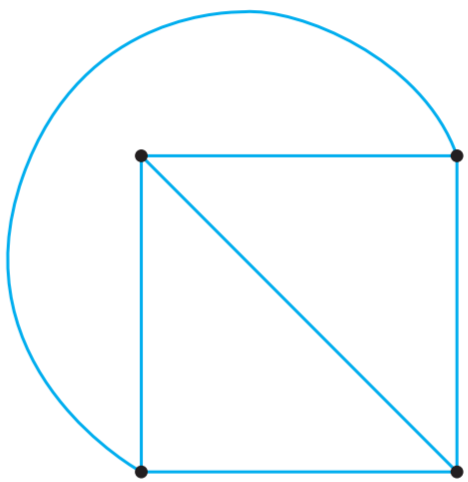
8.1 Introduction

- **병렬 간선**$^{parallel\ edge}$, 예: e_1과 e_2
  같은 정점의 쌍에 결합된 둘 이상의 간선
- **루프**^{loop}, 예: $e_3 = (v_2, v_2)$
  하나의 정점에 결합된 간선
- **고립 정점**$^{isolated\ vertex}$, 예: v_4
  어떠한 간선도 결합되지 않는 정점
- **단순 그래프**$^{simple\ graph}$
  루프도 가지지 않고 병렬 간선도 가지지 않은 그래프

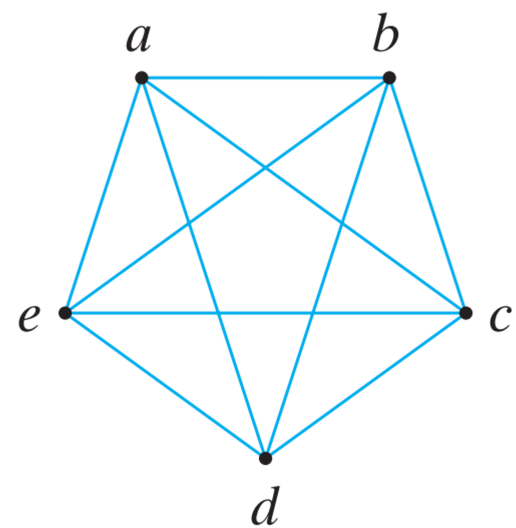


8.1 Introduction

- 정의 8.1.9 n개의 정점을 가지는 **완전 그래프(Complete Graph)**는 서로 다른 모든 정점들 간에 간선이 존재하는 n개의 정점을 가지는 단순 그래프이며, K_n으로 표기한다
- 예제 8.1.10 K_4, K_5



K_4



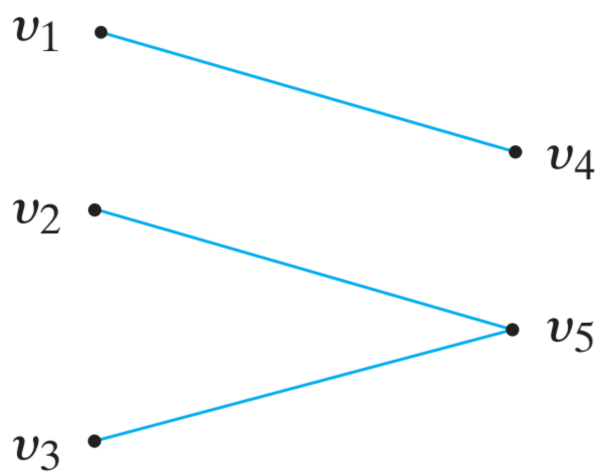
K_5

8.1 Introduction

- 정의 8.1.11 이분그래프(Bipartite Graph) $G = (V, E)$가 이분이라 함은 V의 부분집합 V_1과 V_2가 $V_1 \cap V_2 = \emptyset$, $V_1 \cup V_2 = V$을 만족하며, E의 각 간선이 V_1의 정점과 V_2의정점에 각각 결합된 경우이다.
- 예제 8.1.12 그림의 그래프는 이분이다

$$V_1 = \{v_1, v_2, v_3\}, V_2 = \{v_4, v_5\}$$

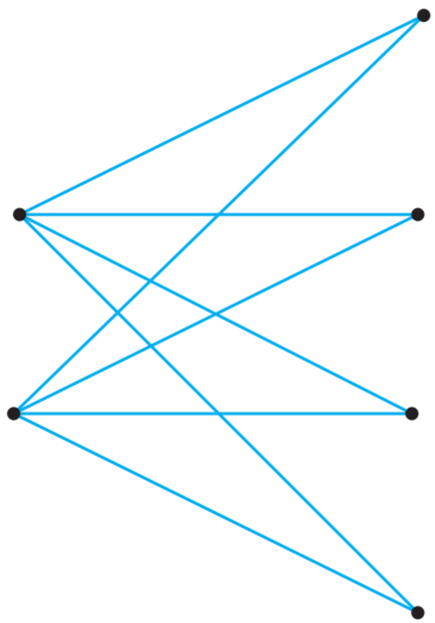
각 간선은 V_1에 속한 정점과 V_2에 속한 정점에 결합된다.



8.1 Introduction

- 정의 8.1.15 $K_{m,n}$이라 표기하는, m개와 n개의 정점을 가지는 완전이분그래프(Complete Bipartite Graph) 는 단순 그래프로 정점의 집합이 m개의 정점을 가지는 집합 V_1과 n개의 정점을 가지는 집합 V_2로 분할되고 간선의 집합은 $\{(v_1, v_2) \mid v_1 \in V_1, v_2 \in V_2\}$이다.

- 예제 8.1.16 $K_{2,4}$.



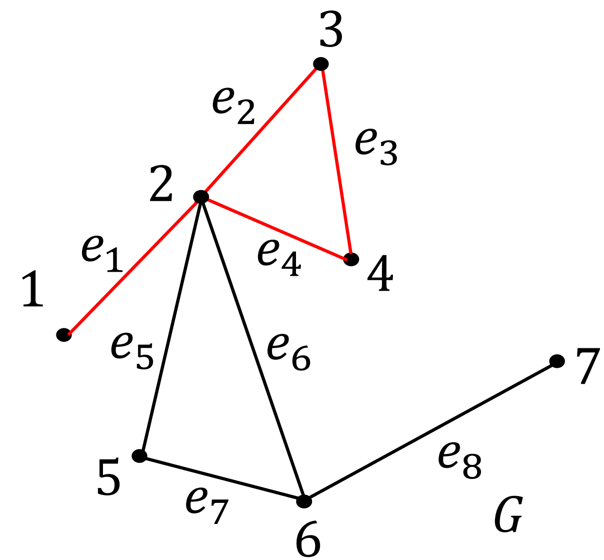
8.2 Paths and Cycles

- 정의 8.2.1 그래프에서 v_0와 v_n을 정점이라 하자. v_0에서 v_n까지의 길이가 n인 **경로**^{path}는 v_0에서 시작하여 v_n에서 끝나는 $n+1$개의 정점과 n개의 간선이 교대로 나타나는 순서열이다.

$$(v_0, e_1, v_1, e_2, v_2, \dots, v_{n-1}, e_n, v_n)$$

$i = 1, \dots, n$에 대해 간선 e_i는 정점 v_{i-1}과 v_i에 결합되어 있다.

- $(1, e_1, 2, e_2, 3, e_3, 4, e_4, 2)$는 정점 1 에서 정점 2로 가는 길이가 4 인 경로이다.
- 경로 (6)은 정점 6에서 정점 6으로 가는 길이가 0 인 경로이다.
- 병렬 간선이 없는 때는 간선없이 경로를 표현 할 수 있다.
예, (1, 2, 3, 4, 2).



8.2 Paths and Cycles

- **정의 8.2.4** 그래프 G의 임의의 정점들 v와 w가 주어졌을 때, v에서 w로의 경로가 존재한다면 그래프 G는 **연결**되어 있다.
- **예제 8.2.5** 그림 2.1(↙)의 그래프 G는 연결되어 있다.
- **예제 8.2.6** 그림 2.2(↘)의 그래프 G는 연결되어 있지 않다. 예, 정점 v_2에서 정점 v_5로 가는 경로가 없기 때문이다.

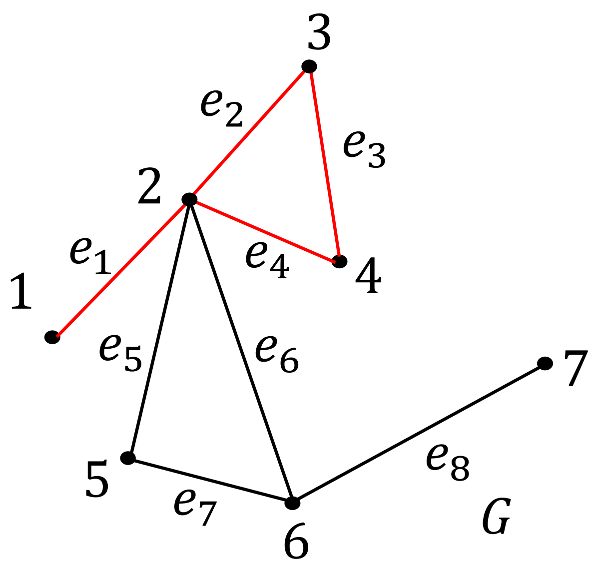


그림 2.1

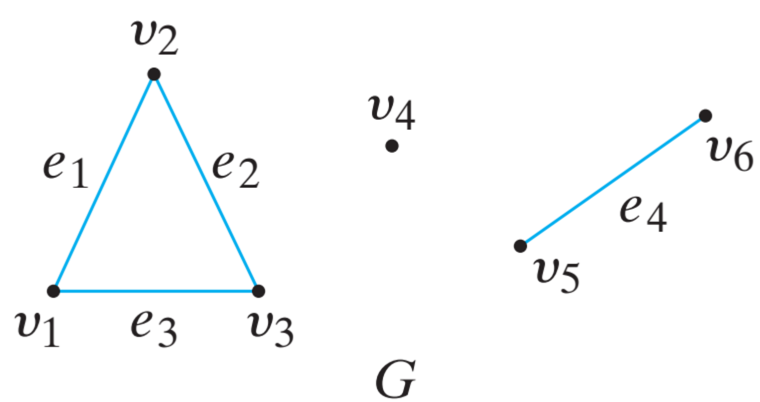
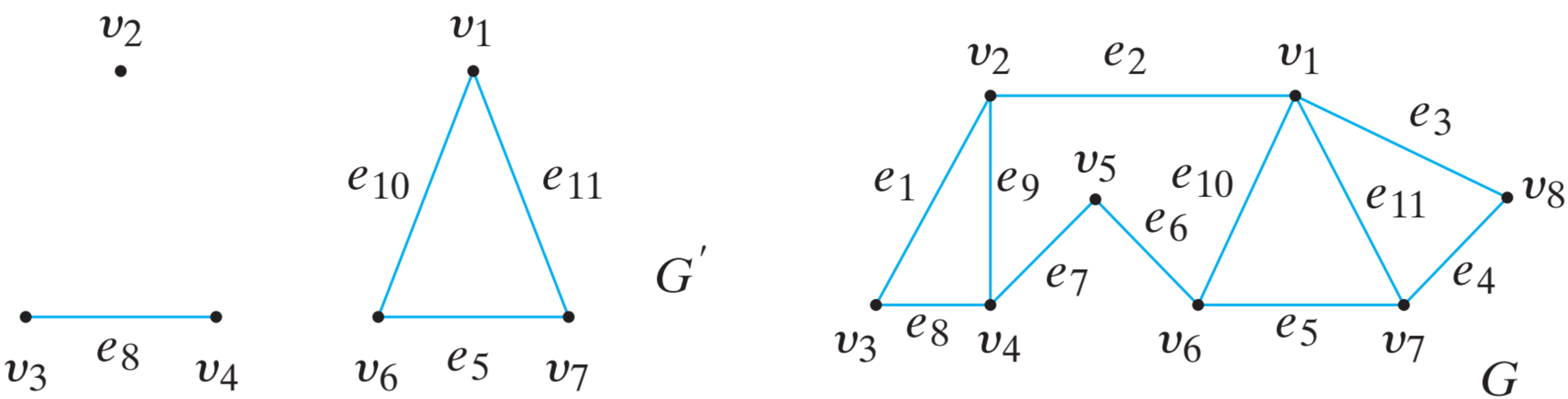


그림 2.2

8.2 Paths and Cycles

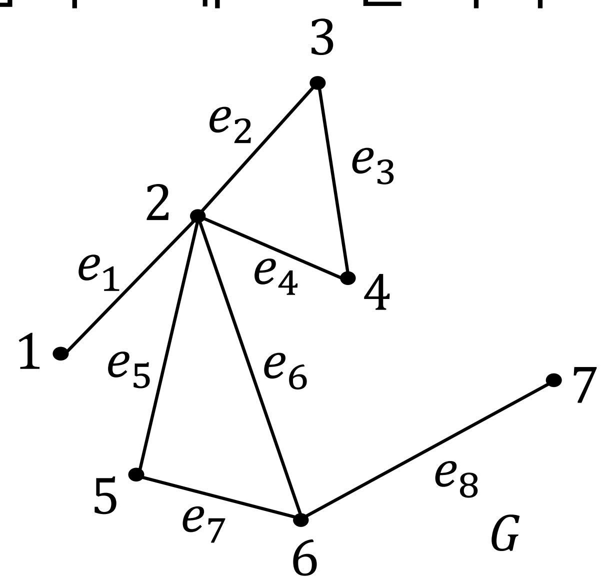
- 정의 8.2.8 $G = (V, E)$를 그래프라 하자. 다음과 같을 때 (V', E')을 G의 부분그래프(Subgraph)라 한다
  (a) $V' \subseteq V$ 이고 $E' \subseteq E$이다
  (b) 모든 간선 $e' \in E'$에 대하여, e'이 v'과 w'에 결합된다면, $v', w' \in V'$이다.

- 예제 8.2.9 그래프 $G' = (V', E')$는 $V' \subseteq V$이고 $E' \subseteq E$이므로 그래프 $G = (V, E)$의 부분 그래프이다



8.2 Paths and Cycles

- 정의 8.2.11 G는 그래프이고 v를 G의 한 정점이라 하자. v에서 시작되는 어떤 경로에 포함된 G안의 모든 간선과 정점으로 구성된 G의 부분그래프 G'을 v를 포함하는 G의 Component (요소)라고 한다.
- 예제 8.2.12 그림의 그래프 G는 하나의 요소를 가진다.



- 그래프가 연결 되어 있을 필요충분조건은 그래프가 정확히 하나의 요소를 가진다.

8.2 Paths and Cycles

□ 예제 8.2.13 그림의 그래프 G는 3개의 요소를 가진다.
v_3를 포함하는 G의 요소는 다음의 부분 그래프이다.

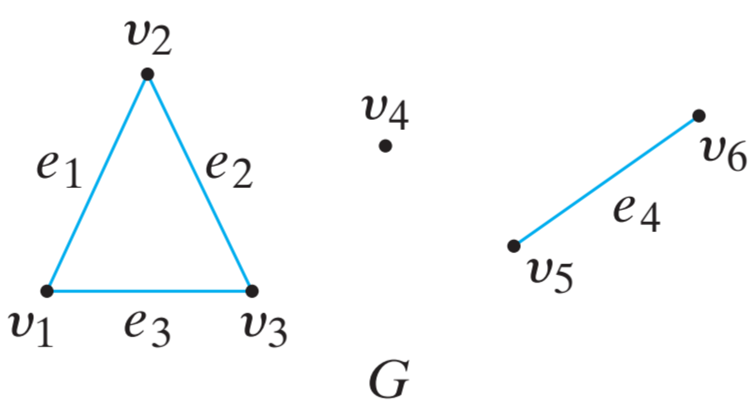
$$G_1 = (V_1, E_1), V_1 = \{v_1, v_2, v_3\}, E_1 = \{e_1, e_2, e_3\}$$

v_4를 포함하는 G의 요소는 다음의 부분 그래프이다.

$$G_2 = (V_2, E_2), V_2 = \{v_4\}, E_2 = \emptyset$$

v_5를 포함하는 G의 요소는 다음의 부분 그래프이다.

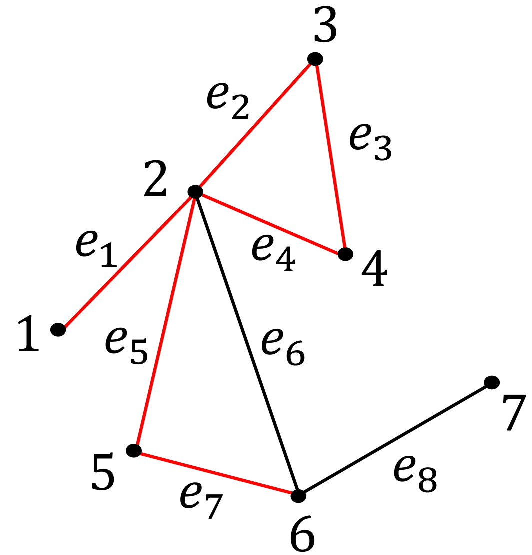
$$G_3 = (V_3, E_3), V_3 = \{v_5, v_6\}, E_3 = \{e_4\}$$



8.2 Paths and Cycles

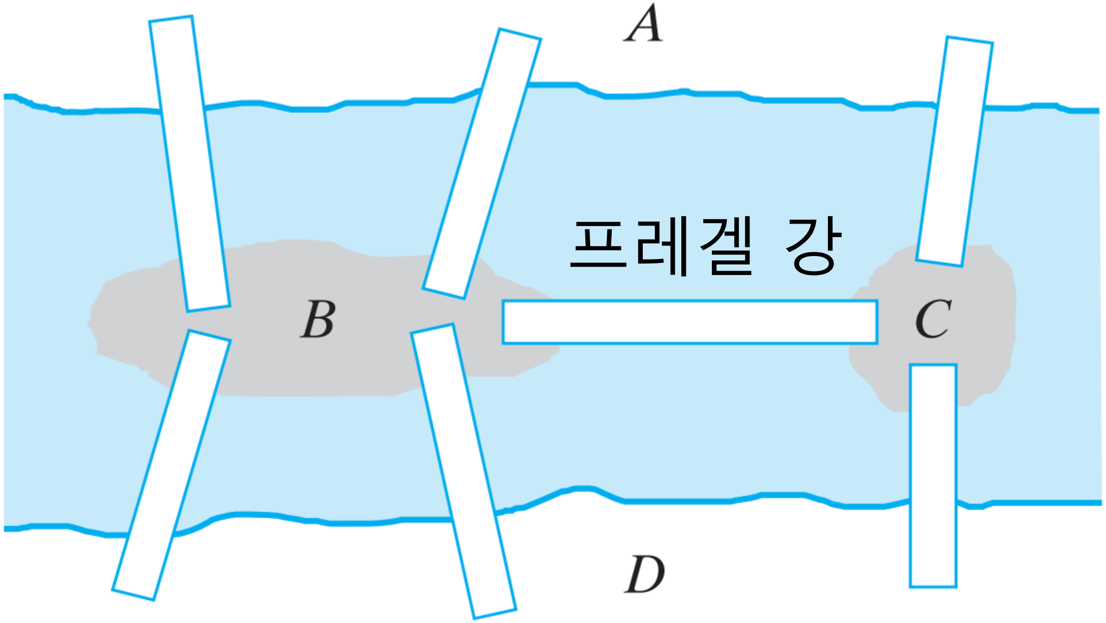
- 정의 8.2.14 v와 w를 그래프 G안의 정점이라 하자.
  - v에서 w로의 **단순 경로**$^{simple\ path}$
    v에서 w까지 반복되는 정점이 없는 경로
  - **사이클**^{cycle} 또는 **회로**^{circuit}
    v에서 v까지 반복되는 간선 없이 길이가 0이 아닌 경로
  - **단순 사이클**$^{simple\ cycle}$
    시작과 끝을 제외하고는 반복되는 정점이 없는 v에서 v로의 사이클

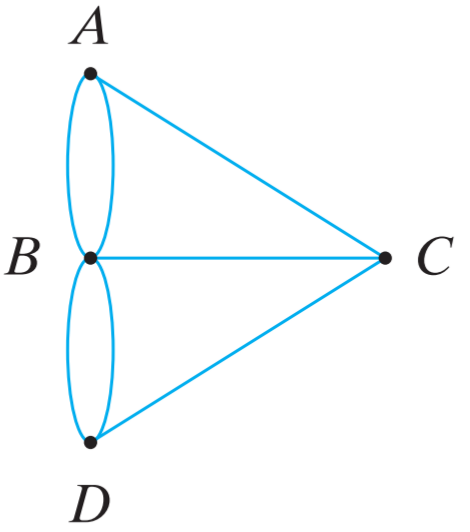
경로	단순 경로?	사이클?	단순 사이클?
(6,5,2,4,3,2,1)	No	No	No
(6,5,2,4)	Yes	No	No
(2,6,5,2,4,3,2)	No	Yes	No
(5,6,2,5)	No	Yes	Yes
(7)	Yes	No	No



8.2 Paths and Cycles

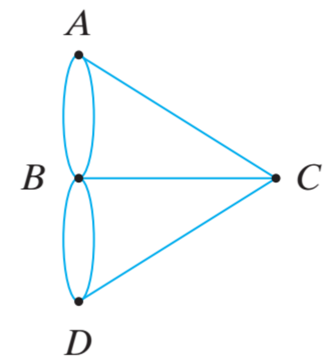
- 강에 있는 2개의 섬이 다리에 의해 각각 연결되어 있고 강둑도 다리로 연결되어 있다. 문제는 어떤 위치 $A, B, C,$ 또는 D에서 출발하여 각각의 다리를 한 번씩만 건너서 출발점으로 돌아오는 것이다.
- 쾨니히스베르크 다리 문제는 그림(↘)의 그래프에서 모든 간선과 모든 정점이 다 포함되어 있는 사이클을 찾는 문제로 변형 된다. 이런 사이클을 오일러 사이클이라 한다.





8.2 Paths and Cycles

- **오일러 사이클 :** 시작 정점 부터 G의 모든 간선을 한번씩만 지나 시작 정점으로 돌아오는 Path
- 정점 v의 **차수**degree $\delta(v)$는 v에 결합된 간선의 수이다. 루프는 2로 카운트 한다.



- 모든 정점의 차수가 k로 동일한 G를 k-Regular 그래프라 한다.
- 정리 8.2.17 그래프 G가 오일러 사이클을 가지면, G는 연결되어 있고 각 정점은 짝수 차수이다.
- 증명 G가 오일러 사이클을 가진다고 가정하자.
  v와 w가 G의 정점이면, v에서 w로 까지 오일러 사이클의 일부는 v에서 w까지의 경로이다. 따라서 G는 연결되어 있다.
  v를 G의 임의의 정점이라 하자.
  어떤 간선으로 v에 도착하면 다른 간선을 따라 반드시 떠나야 한다. 또한 v에 연결된 모든 간선은 반드시 사용되어야 한다.
  짝수개의 간선이 v에 연결되어야 함을 뜻한다.

8.2 Paths and Cycles

- 정리 8.2.18 G가 연결된 그래프이고 모든 정점이 짝수 차수를 가지면 G는 오일러 사이클을 가진다. (정리 8.2.17의 역)
- 예제 8.2.19 G가 그림 8.2.10의 그래프라면 정리 8.2.18을 사용하여 G에 오일러가 사이클이 있는지 찾아라

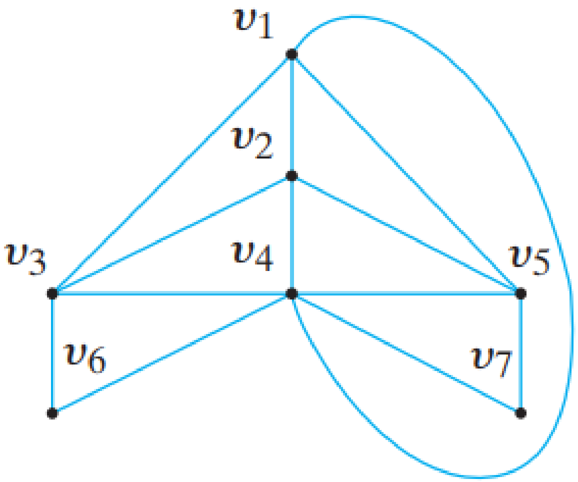


Figure 8.2.10 The graph for Example 8.2.19.

- $\delta(v_1) = \delta(v_2) = \delta(v_3) = \delta(v_5) = 4$
  $\delta(v_6) = \delta(v_7) = 2, \delta(v_4) = 6.$
  모든 정점이 짝수의 차수를 가지므로 다음의 오일러 사이클을 가진다.
  $(v_6, v_4, v_7, v_5, v_1, v_3, v_4, v_1, v_2, v_5, v_4, v_2, v_3, v_6)$

8.2 Paths and Cycles

- 정리 8.2.21 G가 m개의 간선과 정점 $\{v_1, v_2, \dots, v_n\}$을 가진 그래프라면

$$\sum_{i=1}^{n} \delta(v_i) = 2m$$

특히 그래프 안의 모든 정점의 차수의 합은 짝수이다.

- 증명 모든 정점의 차수를 합할 때, 각 간선 (v_i, v_j)를 두 번씩 센다. 한 번은 v_i의 차수에서 (v_i, v_j)로 세고, v_j 의 차수에서 (v_j, v_i)로 한 번 더 센다. 따라서 주어진 결론이 성립한다.

- **오일러 경로** : G의 모든 간선을 지나나 시작점과 끝점은 다를 수 있는 Path (홀수 차수의 정점이 0 또는 2개)

8.2 Paths and Cycles

- 따름 정리 8.2.22 어떤 그래프에서도 홀수 차수의 정점의 수는 짝수 개다.
- 증명 정점을 두 그룹으로 나누자. 짝수 차수를 가지는 것을 $x_1, \ldots, x_m$라 하고, 홀수 차수를 가지는 것을 $y_1, \ldots, y_n$이라 하자.

$$S = \delta(x_1) + \delta(x_2) + \cdots + \delta(x_m),\ T = \delta(y_1) + \delta(y_2) + \cdots + \delta(y_n)$$

정리 8.2.21에 의해, $S + T$는 짝수이다. S가 짝수의 합이므로 S는 짝수이다. 그러므로 T는 짝수이다. 그런데 T가 n개의 홀수의 합이므로 n은 짝수이다.

8.2 Paths and Cycles

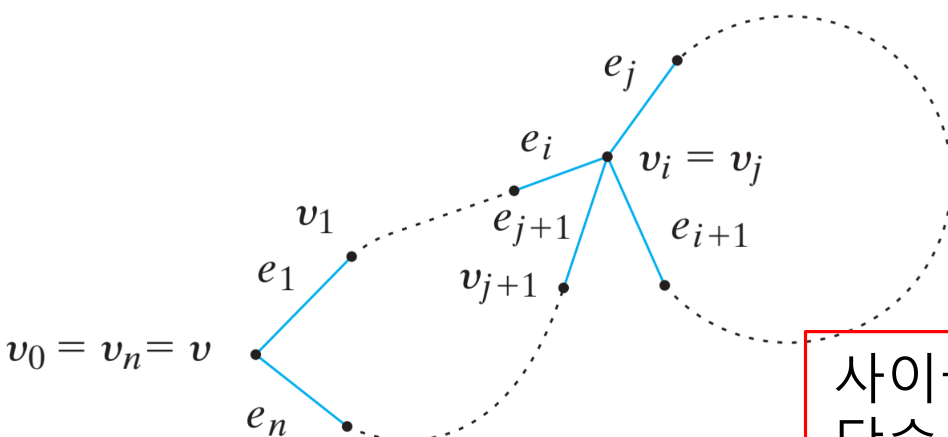
- **오일러 회로와 오일러 경로**
- 그래프에 정확히 두 개의 홀수 정점이 있으면 적어도 하나의 오일러 경로가 있으나 오일러 회로는 없다. 각 오일러 경로는 홀수 정점 중 하나에서 시작되고 다른 하나에서 끝난다.
- 그래프에 모든 짝수 정점이 있으면 적어도 하나의 오일러 회로가 있고(오일러 경로) 오일러 회로는 모든 정점에서 시작하고 끝날 수 있다.
- 그래프에 홀수 정점이 두 개 이상 있으면 오일러회로 또는 오일러 경로가 없다.
- 오일러 회로는 항상 오일러 경로이나 오일러 경로는 오일러 회로가 아닐 수 있다.

8.2 Paths and Cycles

- 정리 8.2.23 어떤 그래프가 모든 정점과 모든 간선을 포함하는 v에서 $w(v \neq w)$로 가는 반복된 간선이 없는 경로 P가 존재한다.
$\Leftrightarrow$ 그 그래프가 연결되고 v와 w 만이 홀수 차수를 가진이다.
- 증명 $\Rightarrow$) 그 그래프는 확실히 연결되어 있다.
v에서 w로의 간선 e를 추가한다면, 그 결과 그래프는 오일러 사이클을 가진다. 정리 8.2.17에 의해 모든 정점은 짝수 차수를 가진다. e을 제거함으로써 v와 w의 차수는 1씩 감소된다. 따라서 원래의 그래프에서 v와 w는 홀수 차수를 가지고, 다른 모든 정점은 짝수 차수를 가진다.
- $\Leftarrow$) v에서 w로의 간선 e를 임시적으로 추가하자. 그 결과 그래프 G'은 연결되고 모든 정점은 짝수 차수이다. 정리 8.2.18에 의해 G'은 오일러 사이클을 가진다. 오일러 사이클에서 e를 제거하면, v에서 w로 가는 모든 정점과 간선을 포함하는 반복되는 간선 없는 경로를 얻을 수 있다.

8.2 Paths and Cycles

- 정리 8.2.24 그래프 G가 v에서 v로의 사이클을 포함하면 G는 v에서 v로의 단순 사이클을 포함한다.
- 증명 $v = v_0 = v_n$이라 할 때 v에서 v로의 사이클 C를
  $C = (v_0, e_1, v_1, \dots, e_i, v_i, e_{i+1}, . . . , e_j, v_j, e_{j+1} v_{j+1}, \dots, e_n, v_n)$ 라하자.
  C가 단순 사이클이 아니라면 $i < j < n$에 대하여 $v_i = v_j$인 정점이 있고, C를 다음처럼 대체 할 수 있다.
  $$C' = (v_0, e_1, v_1, \dots, e_i, v_i, e_{j+1}, v_{j+1}, \dots, e_n, v_n)$$
  C'이 v에서 v로의 단순 사이클이 아니라면 앞과정을 반복한다.
  결국 v에서 v로의 단순 사이클을 얻을 수 있다.



사이클: 간선 반복 없음
단순 사이클: 정점 반복 없음

8.3 Hamiltonian Cycles and the Traveling Salesperson

- **해밀턴 사이클** : 그래프에서 모든 정점을 한 번씩만, 예외로 출발과 도착 정점만 두 번, 포함되는 사이클을 이라 한다
- 예제 8.3.1 그림의 그래프에서 사이클 (a, b, c, d, e, f, g, a)은 Hamiltonian cycle이다. 그러나 그림 8.3.4는 차수가 홀수인 정점이 있어 오일러 사이클을 가지고 있지는 않다.

  Hamiltonian cycle의 유무를 알수 있는 필요충분 조건은 쉽게 확인할 수 없다.

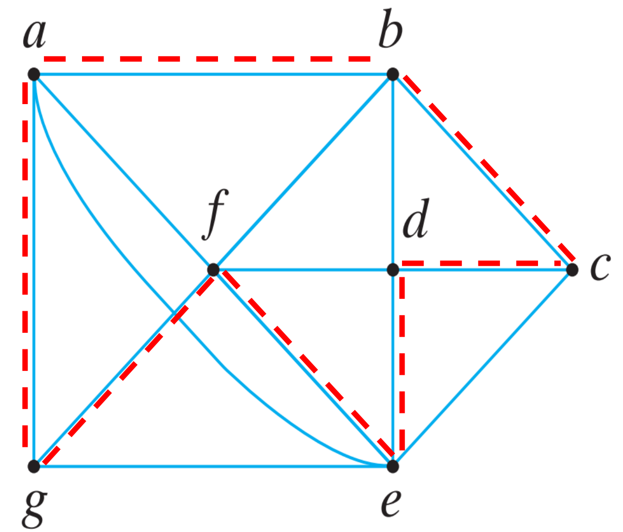


Figure 8.3.4 A graph with a Hamiltonian cycle.

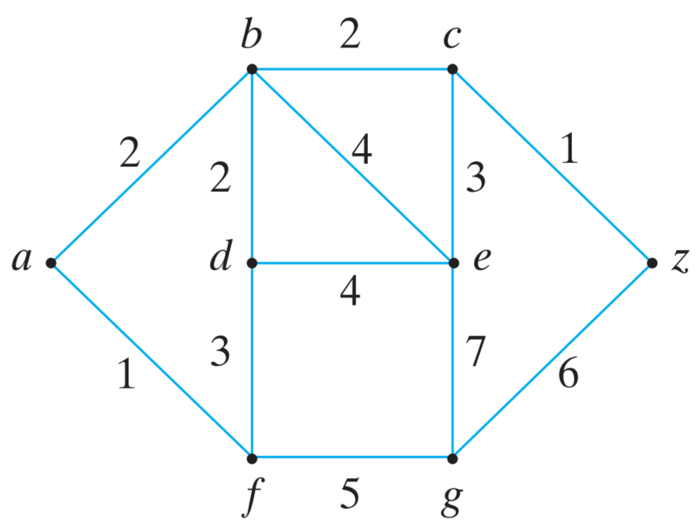
- **해밀턴 경로** : 모든 정점을 한 번씩만 지나는 Path로 Tacerable Path라 한다.

8.3 Hamiltonian Cycles and the Traveling Salesperson

- 판매원 방문 문제traveling salesperson problem는 그래프에서 최소 총 거리의 해밀턴 사이클을 찾는 문제와 관련이 있다.
- 이 문제는 가중치 그래프 G가 주어지면 G 안에서 최소 길이를 가지는 해밀턴 사이클을 찾는 것이다.
- 가중치 그래프에서 도시를 정점으로, 거리를 간선으로 생각한다면, 판매원 방문 문제는 판매원이 어느 도시를 출발하여 각 도시를 한 번 씩 만 방문하여 출발지로 돌아오는 최소의 경로를 찾는 것이다.
- n개의 간선을 가지는 그래프에 대해 실행 시간이 $\Theta(n)$인 오일러 사이클을 찾는 알고리즘은 알려져 있다.
- 실행 시간이 다항식인 해밀턴 사이클을 찾는 알고리즘은 알려지지 않으나 수백개의 정점까지는 최적의 솔루션을 제공하는 Software(Such as Concorde)가 있다

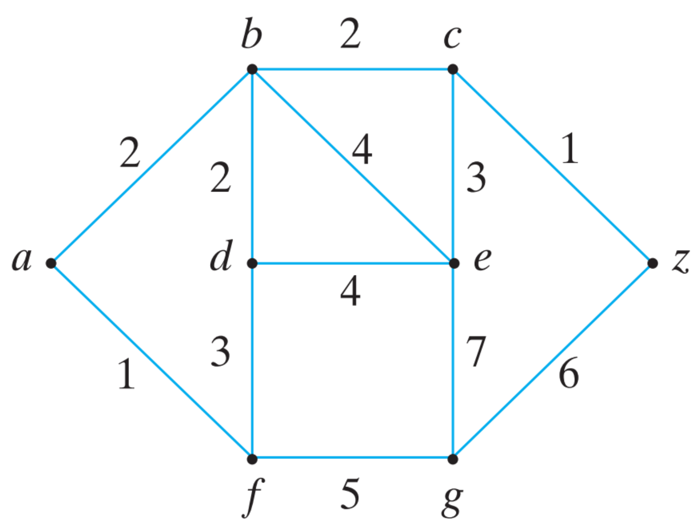
8.4 A Shortest-Path Algorithm

- **가중치 그래프**는 간선에 weight을 부여한 그래프이고, 경로의 길이 는 경로 안의 간선의 가중치의 합이다.
- 간선 (i, j)의 가중치 를 $w(i, j)$로 표기한다.
- 가중치 그래프에서 2개의 정점 간의 최단 경로shortest path ,즉 최소 가중치 경로를 찾는 응용이 많다. 예, 내비게이션, 통신망
- 다익스트라의 최단 경로 알고리즘



8.4 A Shortest-Path Algorithm

- **가중치 그래프**는 간선에 weight을 부여한 그래프이고, 경로의 길이 는 경로 안의 간선의 가중치의 합이다.
- 간선 (i, j)의 가중치 를 $w(i, j)$로 표기한다.
- 가중치 그래프에서 2개의 정점 간의 최단 경로shortest path ,즉 최소 가중치 경로를 찾는 응용이 많다. 예, 내비게이션, 통신망
- 다익스트라의 최단 경로 알고리즘

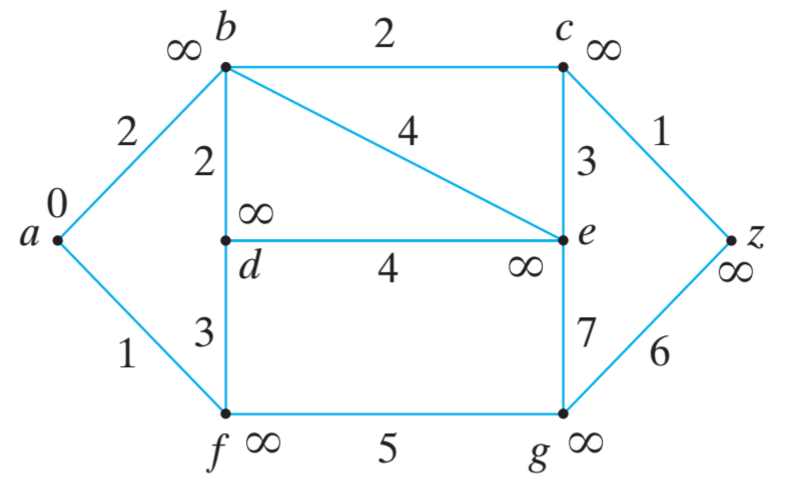


8.4 A Shortest-Path Algorithm

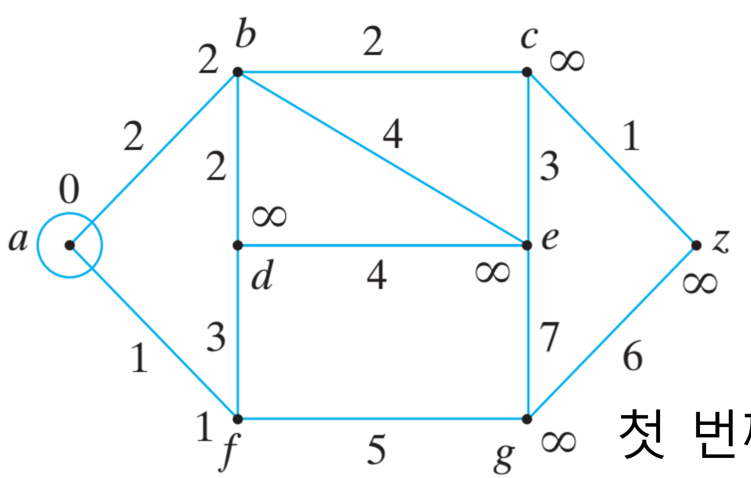
- 정리 8.4.5 n개의 정점으로 구성되고 단순하며 연결된 가중치 그래프에 대하여, 다익스트라의 알고리즘은 최악의 경우 $\Theta(n^2)$ 실행 시간을 가진다.
- 알고리즘 8.4.1 이 알고리즘은 연결된 가중치 그래프에서 정점 a에서 다른 모든 정점 z로의 최단 경로의 길이를 찾는 것이다.
  간선 (i, j)의 가중치는 $w(i, j) > 0$이고
  정점 x의 라벨은 $L(x)$이다 .
  ($L(v)$는 a에서 v까지의 최단 경로의 길이)
  종료 시, $L(z)$는 a에서 z로의 최단경로의 길이이다.
- Input: 연결된 가중치 그래프, 모든 가중치는 양수, 정점 a와 z
- Output: $L(z)$, a에서 z까지의 최단 경로의 길이

8.4 A Shortest-Path Algorithm

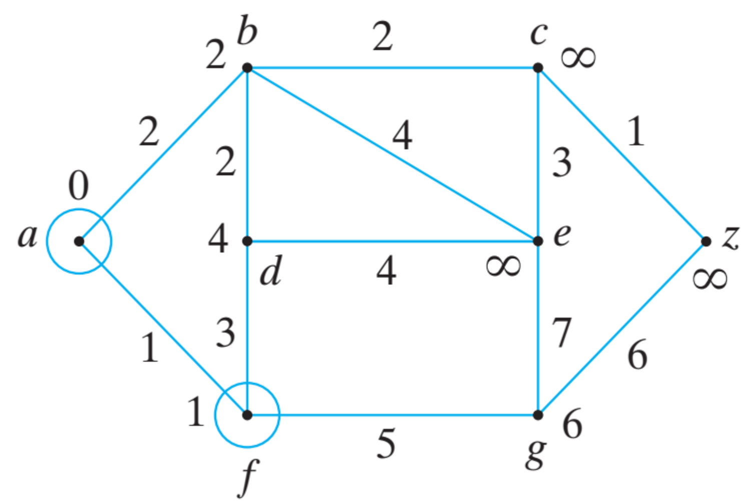
```
dijkstra(w, a, z, L) {
  L(a) = 0
  for all vertices x ≠ a
     L(x) = ∞
  T = set of all vertices
  // T는 a로부터 최단 거리 계산이
  // 아직 안된 정점의 집합
  while (z ∈ T) {
     choose v ∈ T with minimum L(v)
     T = T − {v}
     for each x ∈ T adjacent to v
        L(x) = min{L(x), L(v) + w(v, x)}
  }
}
```



초기화



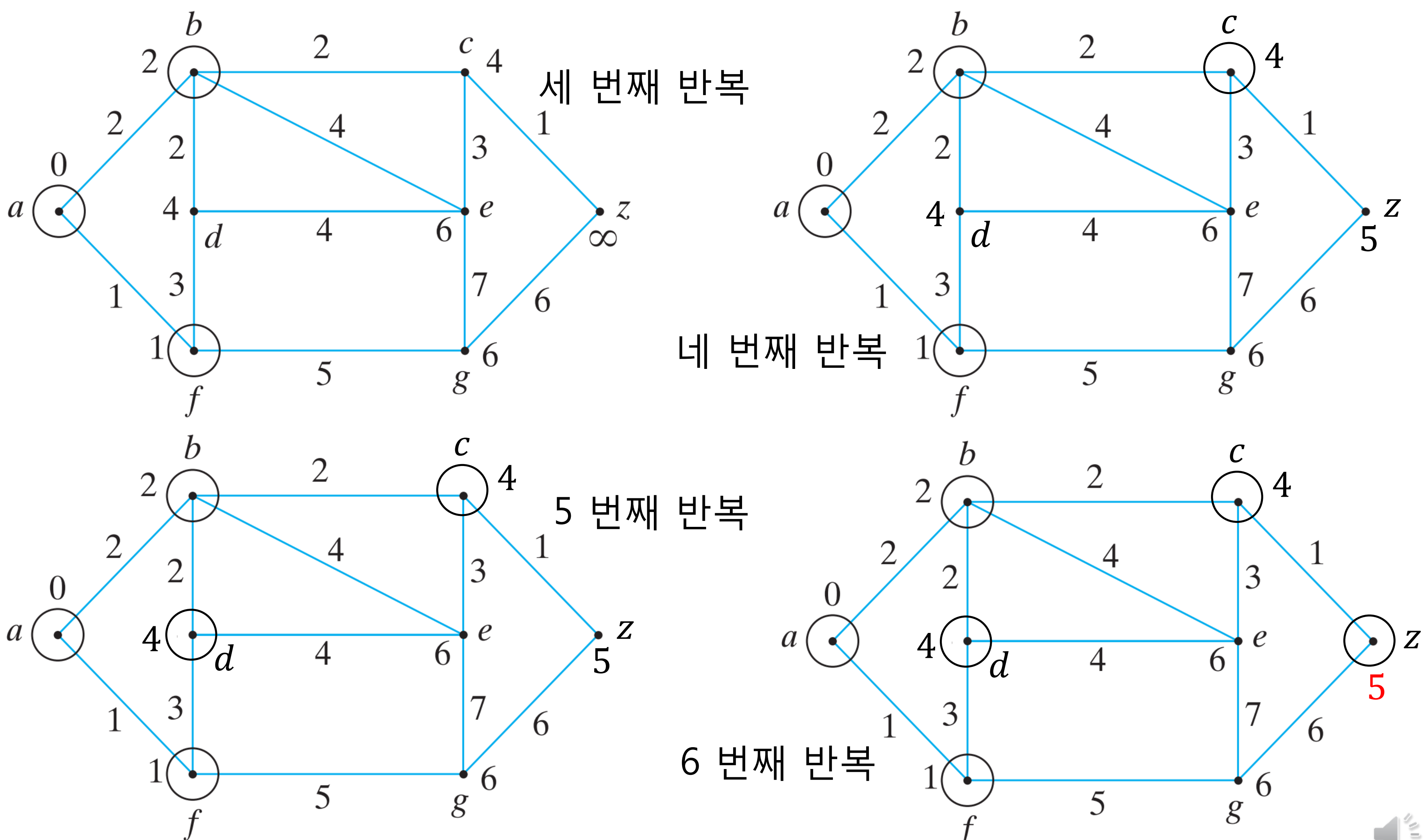
첫 번째 반복



두 번째 반복

8.4 A Shortest-Path Algorithm

예제 8.4.2 다익스트라의 최단 경로 알고리즘



세 번째 반복

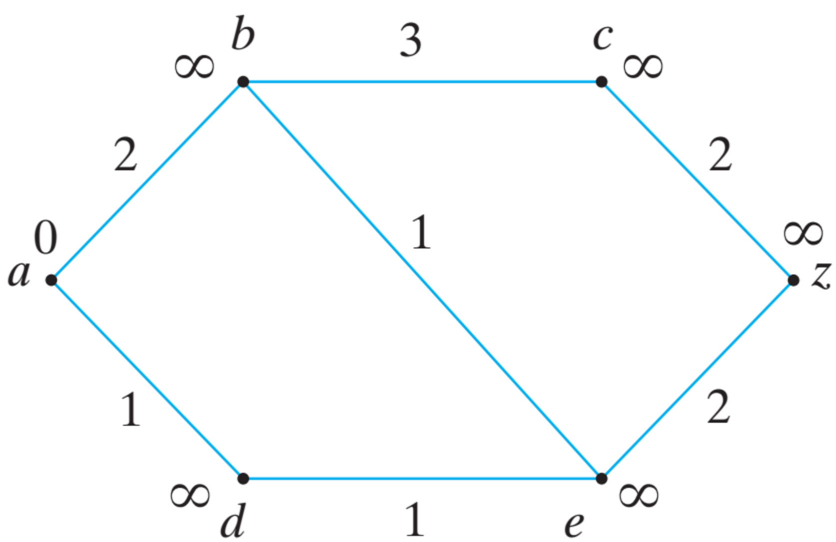
네 번째 반복

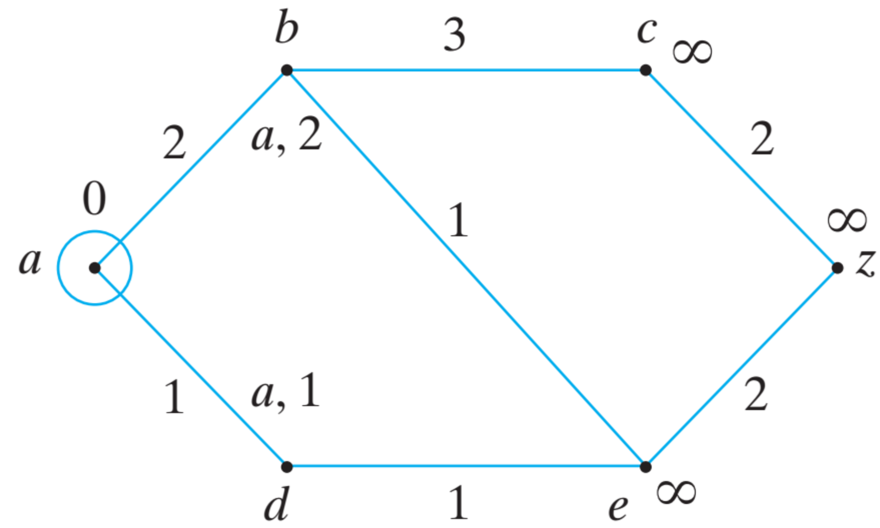
5 번째 반복

6 번째 반복

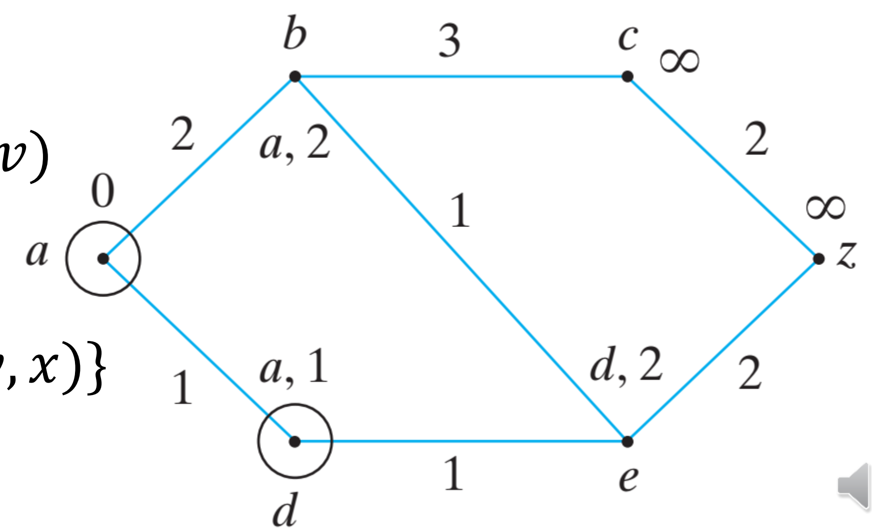
8.4 A Shortest-Path Algorithm

- 예제 8.4.4 a에서 z까지의 최단 경로와 그 길이를 구하라.
- 풀이 어떤 정점에서 왔는 지 표시함



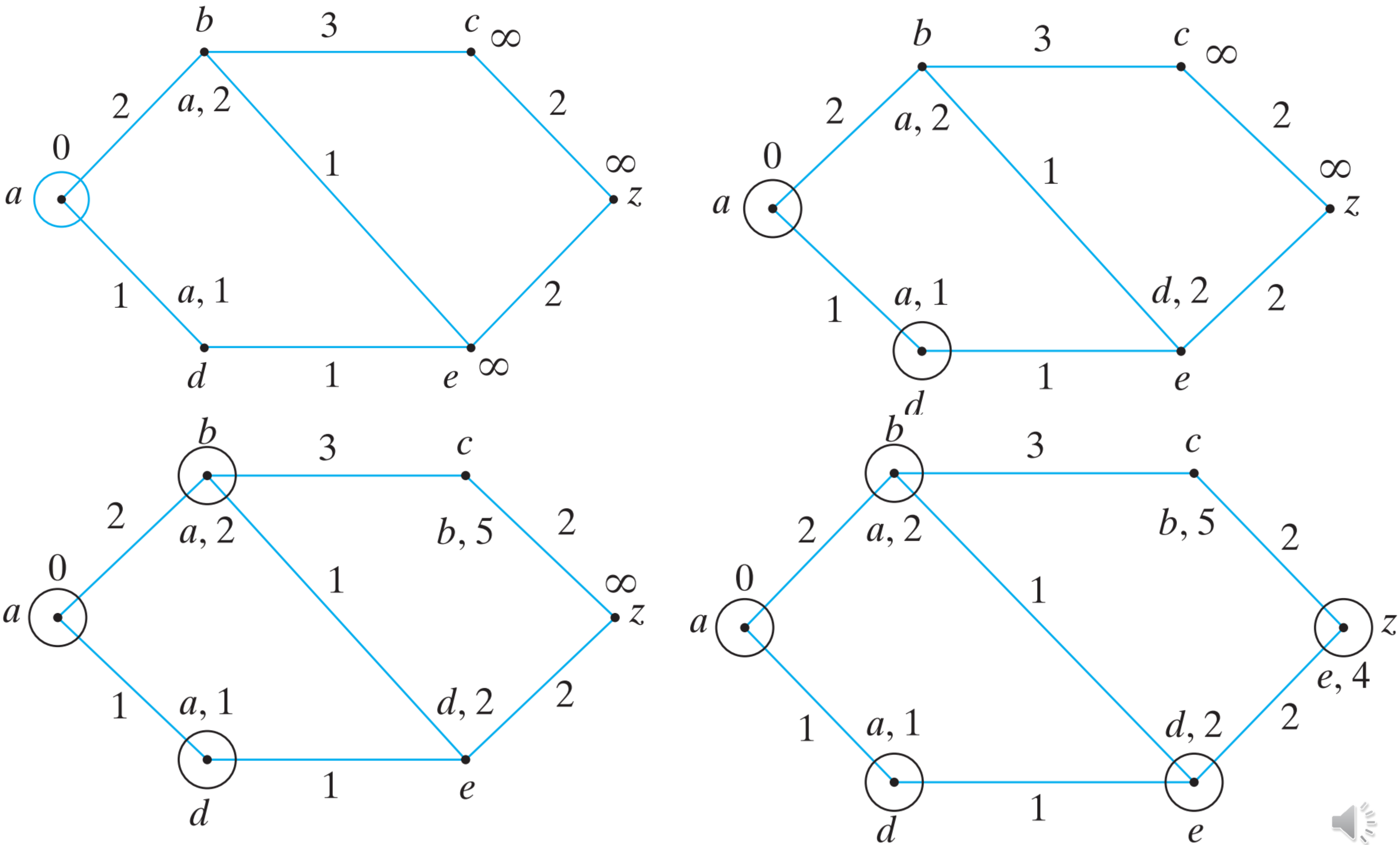


while $(z \in T)$ {
  choose $v \in T$ with minimum $L(v)$
  $T = T - \{v\}$
  for each $x \in T$ adjacent to v
    $L(x) \; = \; \min\{L(x),\; L(v) + w(v,x)\}$
}



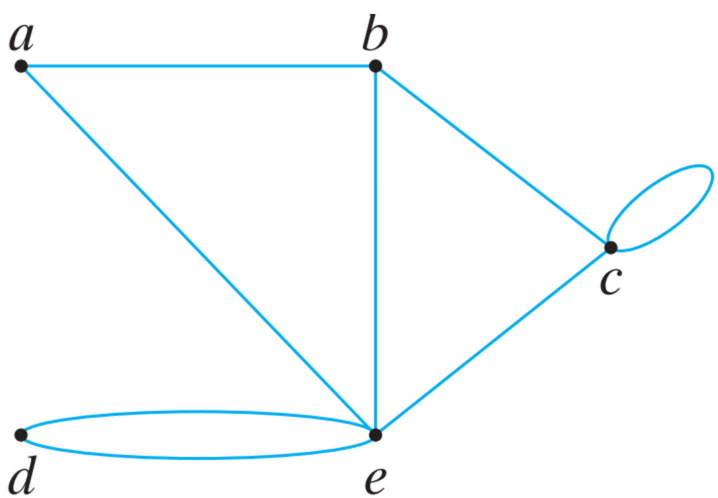
8.4 A Shortest-Path Algorithm

- 최단 경로는 (a, d, e, z)이고 길이는 4이다.



8.5 Representations of Graphs

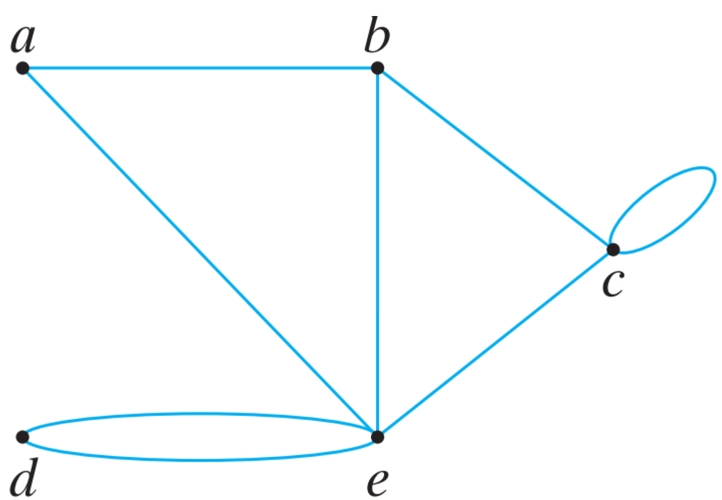
- 예제 8.5.1 인접 행렬Adjacency Matrix
- 정점들의 순서를 정한다. a, b, c, d, e.
- 순서가 매겨진 정점들을 행렬의 가로와 세로에 표기한다.
  - $i \neq j$이면 i행,j열의 성분은 i와j에 결합된 간선의 수이다.
  - $i = j$이면 성분은 i에 루프로 결합되므로 2배가 된다.



$$
\begin{array}{c c}
 & \begin{array}{ccccc} a & b & c & d & e \end{array} \\
\begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} &
\begin{pmatrix}
0 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 1 \\
0 & 1 & 2 & 0 & 1 \\
0 & 0 & 0 & 0 & 2 \\
1 & 1 & 1 & 2 & 0
\end{pmatrix}
\end{array}
$$

8.5 Representations of Graphs

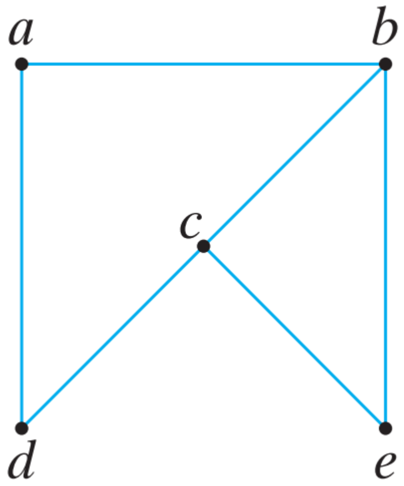
- 그래프 G 안의 정점 v의 차수는 G의 인접 행렬에서 v의 행을 더하거나 또는 v의 열을 더해 얻을 수 있다.
  예, b의 차수는 3
- 인접 행렬은 그래프를 효율적으로 표현하는 방법은 아니다. 행렬이 주 대각선에 대하여 대칭이므로 정보가 대각선을 제외하고는 두 번 나타난다.



$$\begin{array}{c c} & \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} & \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 2 \\ 1 & 1 & 1 & 2 & 0 \end{pmatrix} \end{array}$$

8.5 Representations of Graphs

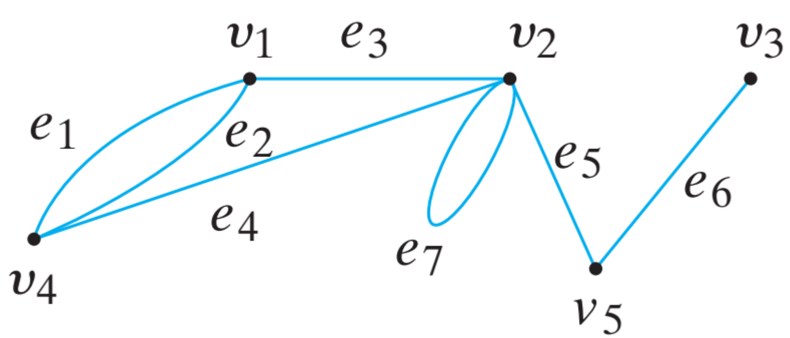
□ 예제 8.5.2 단순그래프의 인접 행렬



$$A = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

8.5 Representations of Graphs

- 예제 8.5.5 결합 행렬
  1. 결합 행렬을 구하기 위해 행에는 정점을, 열에는 간선을 표시한다
  2. v행 e열의 성분이 1 이면 e가 v에 결합되어 있다는 것을 의미하고, 그 외에는 0이다

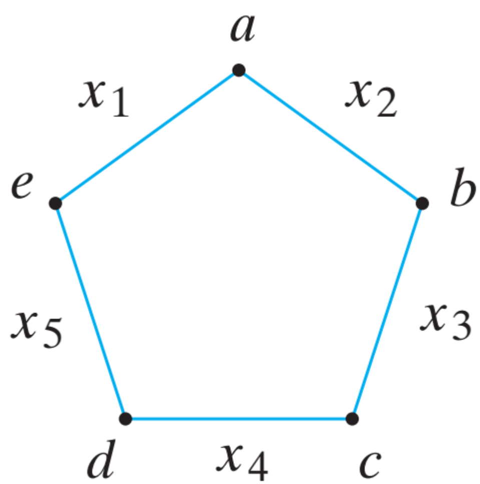


$$
\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array}
\begin{array}{c}
\begin{array}{ccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \end{array} \\
\begin{pmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0
\end{pmatrix}
\end{array}
$$

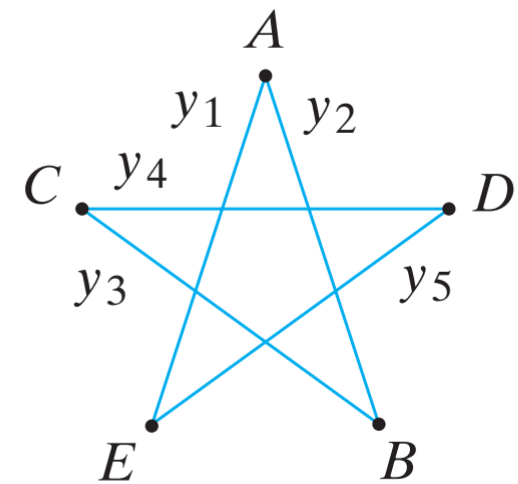
- e_7과 같은 열은 루프로 이해해야 한다
- 루프가 없는 그래프에서 각 열은 2개의 1을 가지고, 행의 합은 그 행에 명시된 정점의 차수를 나타낸다

8.6 Isomorphisms of Graphs

- 동형 그래프



G_1

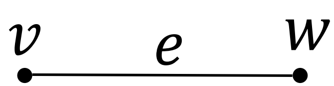


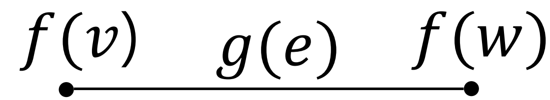
G_2

- $a \to A, b \to B, c \to C, d \to D, e \to E$
  $x_i \to y_i$ for $i = 1,2, \dots, 5$
- 예제 8.6.2 그래프 G_1과 G_2 대한 동형 함수는
  $f(a) = A,\ f(b) = B,\ f(c) = C,\ f(d) = D, f(e) = E,$
  $g(x_i) = y_i, i = 1, \dots, 5.$

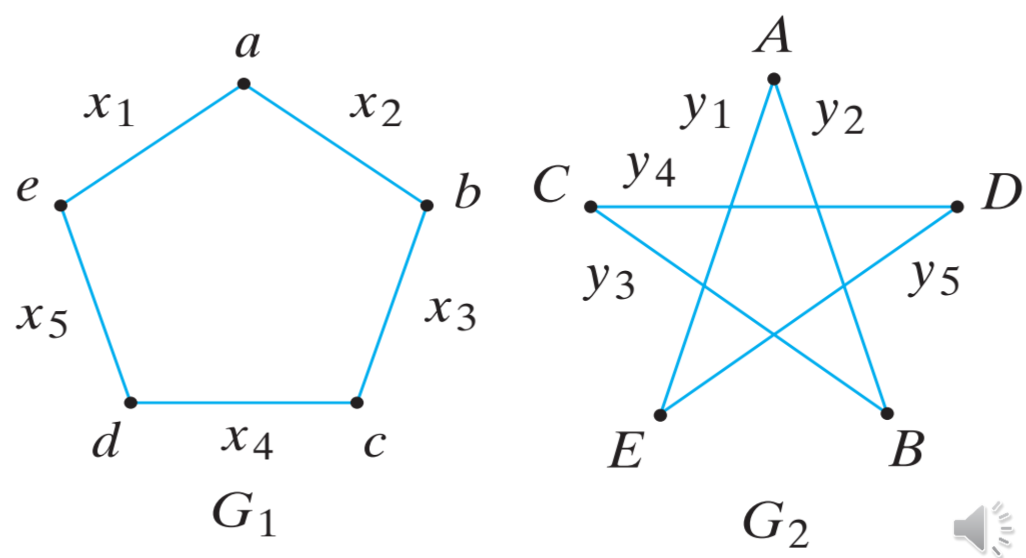
8.6 Isomorphisms of Graphs

- 정의 8.6.1 그래프 $G_1 = (V_1, E_1)$과 $G_2 = (V_2, E_2)$가 **동형**isomorphic
  $f: V_1 \to V_2$가 일대일, 전사 함수이며,
  $g: E_1 \to E_2$가 일대일, 전사 함수이다.
  간선 e는 G_1안의 정점 v와 w에 결합되는 필요충분조건은
  간선 $g(e)$ 는 G_2 안의 정점 $f(v)$와 $f(w)$에 결합된다이다.
- 함수 f와 g의 쌍을 G_2 상에서의 G_1의 **동형 함수**isomorphism이라 한다





- 예제 8.6.2 그래프 G_1과 G_2 대한 동형 함수는 다음과 같다
  $f(a) = A,$
  $f(b) = B,$
  $f(c) = C,$
  $f(d) = D,$
  $f(e) = E,$
  $g(x_i) = y_i, i = 1, \ldots, 5.$

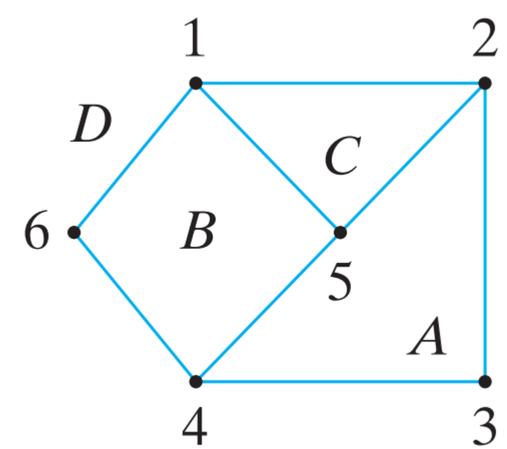


8.7 Planar Graphs

- 정의 8.7.1 간선이 겹치지 않게 평면에 그릴 수 있는 그래프를 평면planar이라 한다.
- 연결된 평면 그래프를 평면에 그리면, 그 평면은 면face 이라는 연속된 영역으로 나누어진다. 면은 그 경계를 형성하는 사이클에 의해 규정된다.
- 정점의 수가 v, 간선의 수가 e, 면의 수가 f인 연결된 평면 그래프에서는 다음식을 만족한다.

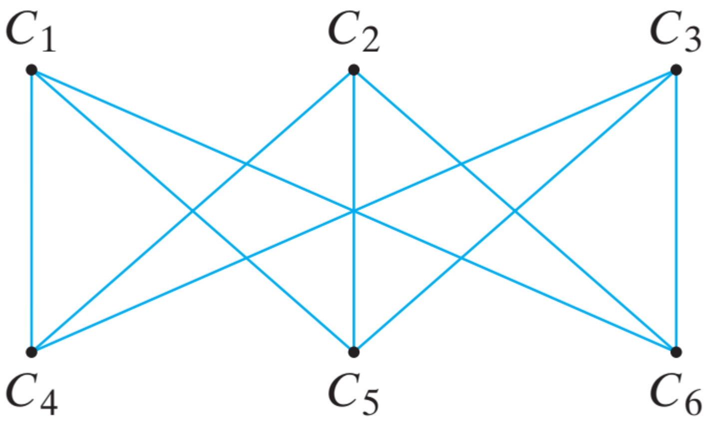
$$f = e - v + 2 \qquad (7.1)$$

- 예: 면 A는 사이클 $(5, 2, 3, 4, 5)$,
  면 C는 사이클 $(1, 2, 5, 1)$에 의해 경계를 이룬다.
  바깥 면 D는 사이클 $(1, 2, 3, 4, 6, 1)$에 의해 경계를 이룬다고 간주한다.
  $f = 4, e = 8, v = 6.$



8.7 Planar Graphs

- 예제 8.7.2 그래프 $K_{3,3}$가 평면 그래프가 아님을 보이라.
- 풀이 $K_{3,3}$가 평면 그래프라고 가정하자.
  모든 사이클이 적어도 4개의 간선을 가지므로 각 면은 적어도 4개의 간선에 의해 경계를 이룬다. 따라서 면의 경계를 이루는 간선의 수 (n)는 적어도 $4f$이다. $(n \geq 4f)$
  평면 그래프에서 각 간선은 경계선을 이루는 사이클에 두 번 나타난다.
  그러므로 $2e \geq 4f$ 이다.



- 식 (7.1)에 의해

$$2e \geq 4(e - v + 2)$$

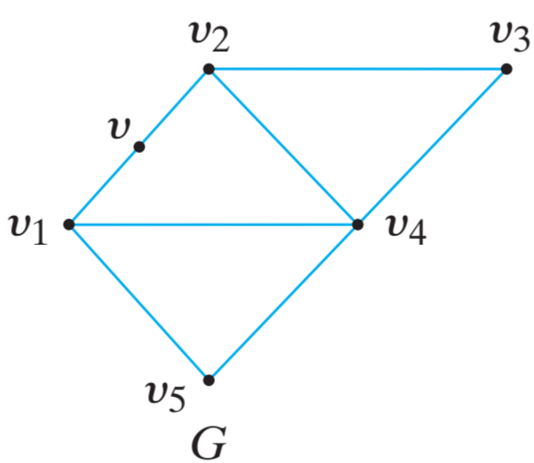
  그림의 그래프에 대해
  $e = 9, v = 6$이므로
  $18 = 2 \cdot 9 \geq 4(9 - 6 + 2) = 20$ 이다.

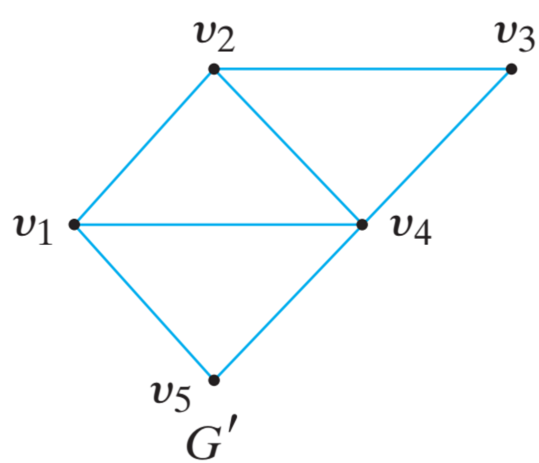
  모순!!

  그러므로 $K_{3,3}$는 평면 그래프가 아니다.

8.7 Planar Graphs

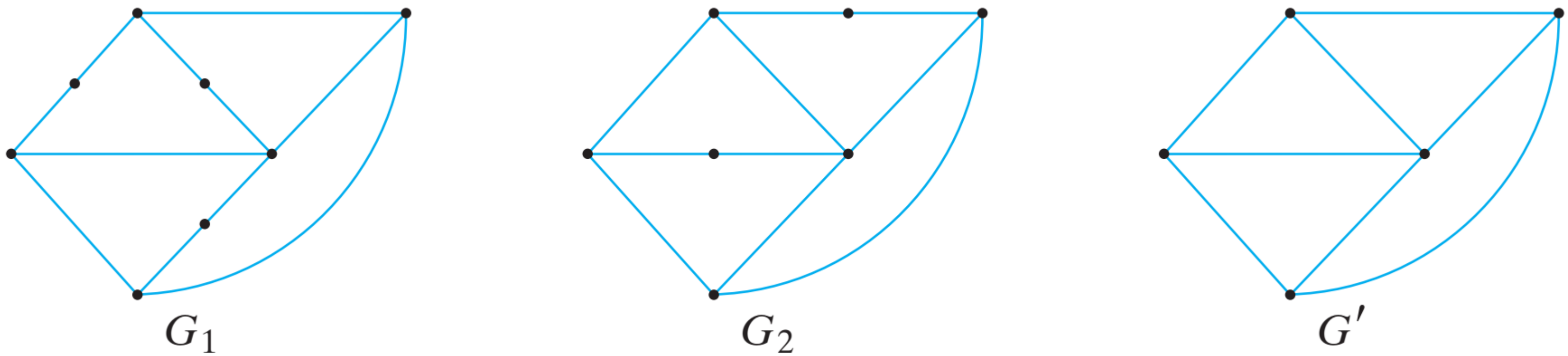
□ 정의 8.7.3 그래프 G가 차수가 2인 정점 v와 $v_1 \neq v_2$인 (v, v_1), (v, v_2)를 가진다면, 간선 (v, v_1)과 (v, v_2)는 **시리즈**series 안에 있다고 말한다.
시리즈 축소series reduction는 그래프 G에서 정점 v를 제거하는 것과 간선 (v, v_1)과 (v, v_2)가 간선 (v_1, v_2)로 대체되는 것으로 구성된다.
그 결과로 나타나는 그래프 G' 은 G로부터 시리즈 축소에 의해 얻을 수 있다고 말하고, 편의상 G는 그 자체로부터 시리즈 축소에 의해 얻을 수 있다.





8.7 Planar Graphs(동상)

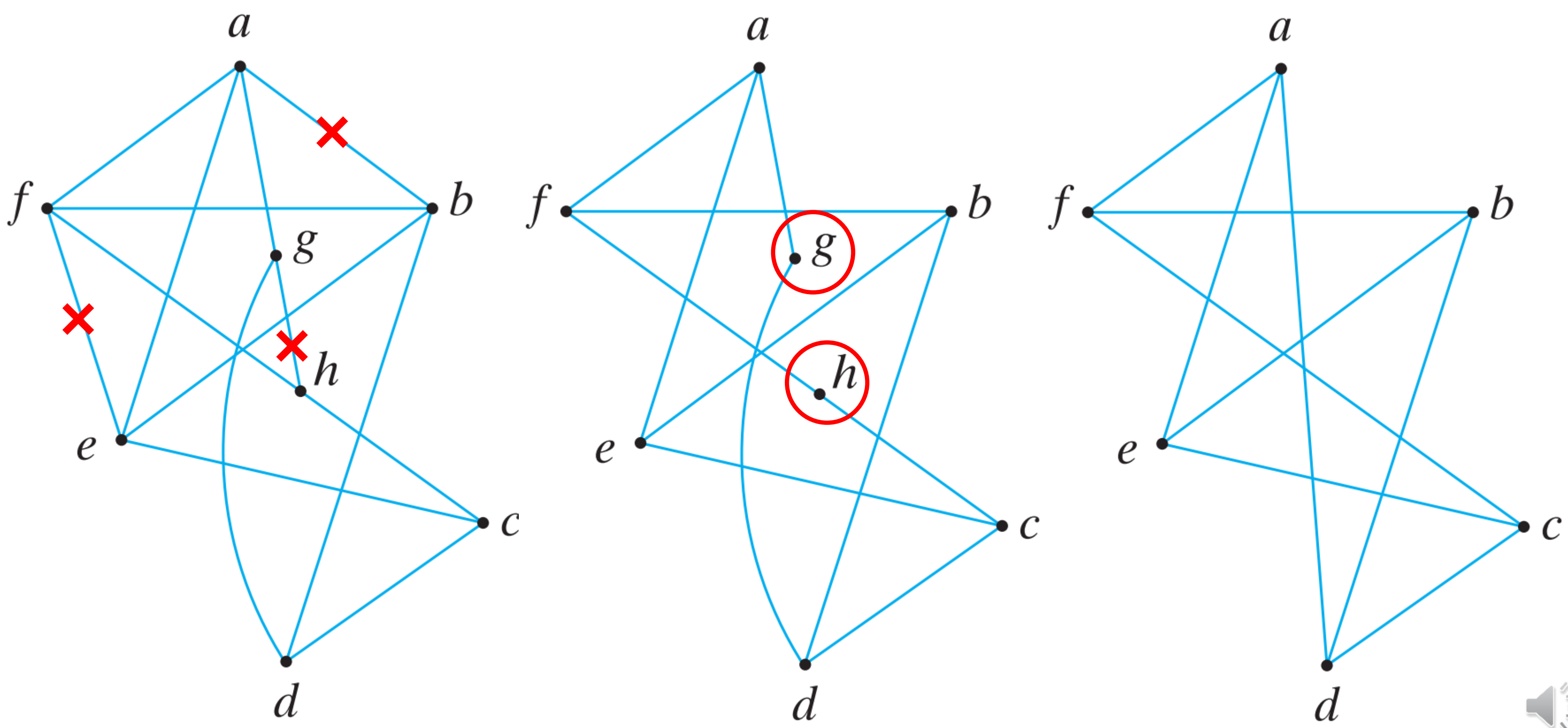
- 정의 8.7.5 G_1과 G_2가 일련의 시리즈 축소를 통해 동형인 그래프로 축소될 수 있을 때, 그래프 G_1과 G_2는 **동상**homeomorphic이라고 한다.
- 예제 8.7.6 그래프 G_1과 G_2는 동상이다. 둘 다 모두 일련의 시리즈 축소에 의해 그래프 G' 으로 축소될 수 있기 때문이다.



- 정의 8.7.3 과 8.7.5에 따라 어떤 그래프도 그 자신의 그래프와 동상이다. 또한 그래프 G_1는 동형인 그래프 G_2로 축소될 수 있을 때, 또는 G_2가 동형인 그래프 G_1으로 축소될 수 있을 때, 그래프 G_1과 G_2는 동상이다.

8.7 Planar Graphs(Kuratowski's Theorem)

- 정리 8.7.7 Kuratowski's Theorem : 어떤 그래프 G가 평면 그래프일 필요충분조건은 G는 K_5 또는 $K_{3,3}$에 동상인 부분 그래프를 포함하지 않는다는 것이다
- 예제 8.7.8 그래프 G가 평면 그래프가 아님을 보이라.
  풀이 간선 제거와 시리즈 축소하여 $K_{3,3}$를 얻는다.



8.7 Planar Graphs(그래프를 위한 오일러 공식)

- 정리 8.7.9 G가 e개의 간선, v개의 정점, f개의 면으로 구성된 연결된 평면 그래프이면 다음의 식이 성립한다.

$$f = e - v + 2 \qquad (7.3)$$

- 증명 간선의 수에 대한 귀납 방법을 사용한다. $e = 1$이라고 가정하자. 그러면 G는 그림의 두 그래프 중 하나이다.

$f = 1, e = 1, v = 2$

$f = 2, e = 1, v = 1$

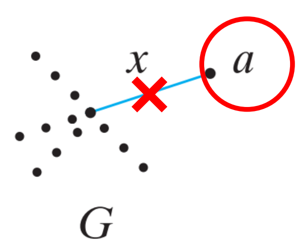
양쪽의 어느 경우라도 이 공식은 성립한다. 기본단계의 증명이 끝났다.

이 공식이 n개의 간선을 가지는 연결된 평면 그래프에서 성립한다고 가정하자.

G가 $n + 1$개의 간선을 가진다고 하자. G가 사이클을 포함하거나 하지 않는다

8.7 Planar Graphs

- G가 사이클이 없다고 가정하자.
  정점 v를 선택하고, v에서 출발하여 하나의 경로를 추적하자.
  G가 사이클이 없으므로 간선을 추적할 때마다 새로운 정점에 도달할 수 있다. 결국 차수가 1 인 어떤 정점 a에 도착할 것이고, 이 정점에서 나갈 수 없다.
  a를 제거하고 a에 결합된 간선 x를 제거 하여 그래프 G'를 얻는다.
  G'는 n개의 간선을 가지고 연결되어 있다. 수학적 귀납법 가정에 의해 G' 에 대해 (7.3)은 성립한다.
  G는 G' 보다 간선이 하나 더 많고, 정점도 하나 더 많으며, 면의 수는 같으므로, 식 (7.3)은 G에 대하서도 성립한다.



8.7 Planar Graphs

- G가 사이클을 포함한다고 가정하자.
  x를 사이클 안의 한 간선이라 하자. x는 두 면의 경계의 한 부분이다. 정점의 제거 없이 간선 x를 제거하여 그래프 G'을 얻는다. G'은 n개의 간선을 가지고 연결되어 있다. 수학적 귀납법 가정에 의해 G' 에 대해 (7.3)은 성립한다.
  G가 G' 보다 하나의 면과 하나의 간선을 더 갖지만 정점의 개수는 같으므로, G에 대해 (7.3)은 역시 성립한다.
- 귀납 단계도 증명되었으므로, 수학적 귀납 원리에 의해 이 정리는 증명된다.

$$f = e - v + 2 \qquad (7.3)$$