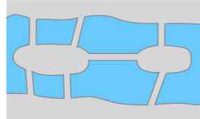


## 7.1 그래프의 기본 개념

교과목명	이산수학	분반		담당교수	김 외 현
학부(과)		학번		성명	

### 참고 그래프 이론의 대표적인 예

퀴니히스베르크 다리 문제  
두 개의 섬과 강둑 사이를 연결하는  
7개의 다리가 있을 때  
각 다리를 꼭 한 번씩만  
건너는 경로를 찾는 문제



### 정의 그래프

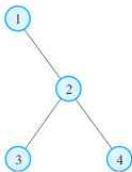
$$G = (V, E)$$

이때  $V = \{v_1, v_2, \dots, v_n\}$

:  $G$ 의 정점(노드)의 집합

$$E = \{(v_i, v_j) \mid (v_i, v_j) \text{는 } v_i \text{와 } v_j \text{를 연결한 선}\}$$

: 정점들의 쌍 즉, 연결선(edge)들의 집합



### 정의 그래프의 2가지 종류

(1) 방향(유향) 그래프

간선에 방향이 존재하는 그래프

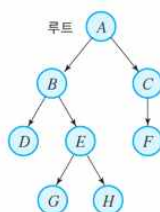
연결선을 화살표로 표시하여 방향을 나타냄

(2) 무향 그래프

간선에 방향이 존재하지 않는 그래프

### 정의 트리

- 사이클(cycle)이 존재하지 않는 그래프임
- 루트(root)라 불리는 특별한 노드가 한 개 존재하고 루트로부터 다른 모든 노드로 가는 경로가 항상 유일하게 존재함
- 루트로 들어오는 연결선이 없으므로 루트는 모든 트리의 출발점이 됨



### 예제

1.  $V = \{1, 2, 3, 4\}$ 이고

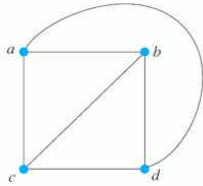
$E = \{(1, 2), (1, 3), (1, 4), (2, 3), (2, 4), (3, 4)\}$ 인  
그래프  $G = (V, E)$ 를 무향 그래프와 유향 그래프로  
나타내보자.

## 7.2 그래프의 용어

교과목명	이산수학	분반		담당교수	김 외 현
학부(과)		학번		성명	

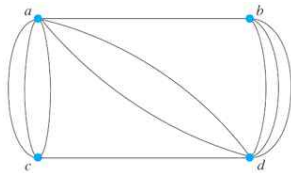
### 정의 단순 그래프

한 쌍의 정점 사이에 많아도 하나의 연결선으로 이루어진 그래프,  
자기 자신으로의 연결선이 없는 그래프



### 정의 멀티 그래프

단순 그래프의 확장으로서 한 쌍의 정점 사이에 연결선의 개수 제한이 없는 그래프



### 정의 연결선

$G = (V, E)$ : 그래프

$\Rightarrow E$ : 그래프의 연결선

if  $(u, v) \in E$ ,

$e$ :  $u$ 와  $v$ 를 연결하는 연결선

$\Rightarrow e$ 는  $u$ 와  $v$ 에 ‘접했다’

$u$ 와  $v$ 를 서로 ‘인접했다’

### 정의 차수

$G = (V, E)$ : 그래프,

$v \in V$ : 정점

$\Rightarrow \deg(v) = (v$ 에 인접하는 연결선들의 개수)  
:  $v$ 의 차수

### 정의 부분 그래프

$G = (V, E)$ ,  $G' = (V', E')$ : 그래프

(1)  $V' \subseteq V$ ,  $E' \subseteq E$

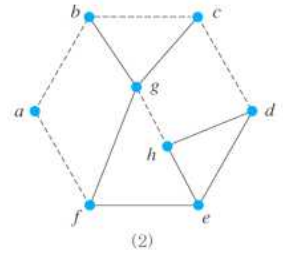
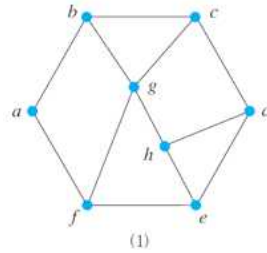
$\Rightarrow G'$ :  $G$ 의 부분 그래프

(2)  $V' = V$ ,  $E' \subset E$

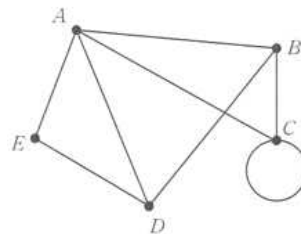
$\Rightarrow G'$ :  $G$ 의 생성 부분 그래프

### 예제

2. 다음 두 그래프를 살펴보자.



3. 다음 그래프에서 각 정점의 차수를 구해보자.



## 정의 그래프에서의 경로

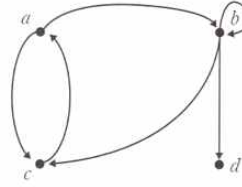
- (1) 경로
  - $v_1$ 에서  $v_n$ 으로 가는 경로  
 $\Rightarrow (v_1, v_2), (v_2, v_3), \dots, (v_{n-1}, v_n) \in E$   
 $v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_n$ 로 표기
  - 경로의 길이: 경로에 있는 연결선의 개수
- (2) 단순 경로  
 경로가 같은 연결선을 두 번 포함하지 않는 경로
- (3) 기본 경로  
 어떤 정점들도 두 번 만나지 않는 경로
- (4) 사이클  
 경로  $v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_n$ 에서  $v_1 = v_n$ 인 경우
- (5) 단순 사이클  
 같은 연결선을 반복하여 방문하지 않는 사이클
- (6) 기본 사이클  
 시작점을 제외한 어떠한 정점도 반복하여 방문하지 않는 사이클

## 정의 연결

- (1) 연결 그래프  
 그래프의 모든 정점들이 연결되어있는 그래프
- (2) 강한 연결 그래프  
 $G = (V, E)$ : 방향 그래프  
 $\forall u, v \in V$ ,  $u$ 에서  $v$ 로의 경로와  
 $v$ 에서  $u$ 로의 경로가 존재하는 방향그래프
- (3) 연결 요소
  - 그래프에서 모든 정점들이 연결되어 있는 부분
  - 연결수: 그래프에서 연결 요소의 개수

## 예제

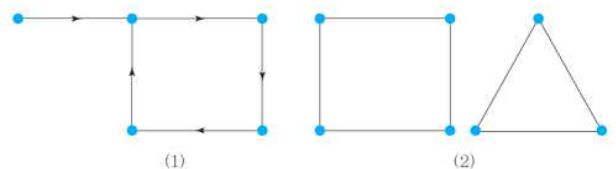
4. 다음 유향 그래프에서 정점  $a$ 에서 정점  $c$ 로 가는 경로를 구하고 사이클을 구해보자.



5. 버스나 지하철의 노선 안내도는 일종의 무향 그래프이다. 다음의 부분 지하철 노선도에서 수원에서 신창까지의 경로가 단순경로인지 아닌지, 사이클인지 확인해보자.



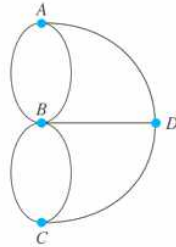
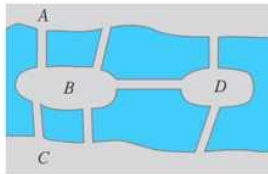
6. 다음 그림이 연결 그래프인지 강한 연결 그래프인지를 판단해보자.



## 정의 멀티 그래프

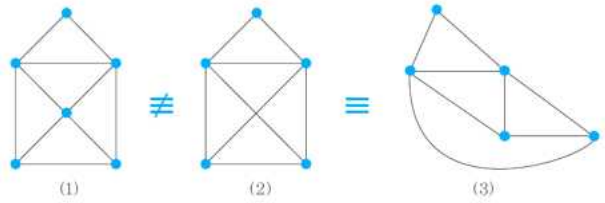
한 쌍의 정점 사이에 연결선의 개수 제한이 없는 중복된 연결선을 허용하는 특별한 그래프

- 쾨니히스베르크 다리 문제를 해결하는 방법은 멀티 그래프를 모델링하는 것임



- 쾨니히스베르크 다리를 그래프로 나타낸  $A$ ,  $B$ ,  $C$ ,  $D$ 로 이름 붙인 점들은 정점을 나타내고, 정점들 사이의 선들은 연결선을 나타냄
- 멀티그래프에서 모든 연결선들을 꼭 한 번씩만 통과하는 경로를 오일러경로라고 하는데, 쾨니히스베르크 다리 문제는 오일러경로를 찾을 수 있는지 여부와 동치임
- 멀티 그래프에서의 오일러경로를 판별하는 규칙은 모든 정점에서 그것과 연결된 연결선의 개수가 홀수인 정점(홀수점)의 개수가 0 또는 2개인 경우임
- 그래프에서는  $A$ ,  $B$ ,  $C$ ,  $D$  4개의 정점들이 모두 홀수점을 가지므로(총 4개) 오일러경로가 없다. 따라서 각 다리를 꼭 한 번씩만 건너는 경로가 존재하지 않음
- 정점들과 연결선으로 이루어진 도형에 대한 초기의 그래프 이론은 단순하고 직관적인 문제 해결에 국한됨
- 복잡한 모델에 대한 이론적인 연구로 발전
- 기하학에 위상적인 개념을 집합시킨 응용으로서 위상 기하학적인 관계를 나타냄
- 그래프에서 점의 위치나 선의 길이 등에는 특별한 의미를 부여하지 않고 위상적인 형태에만 중점 둠

- 아래 그림 (1), (2)는 위상적으로 서로 다르지만, (2)와 (3)은 위상적으로 같음



## 7.3 그래프의 표현

교과목명	이산수학	분반		담당교수	김 외 현
학부(과)		학번		성명	

### 참고 그래프의 표현 방법

- 그래프는 그림을 이용하여 표현하는 것으로 가장 자연스럽게 이해하기 쉬운 방법임
- 컴퓨터는 그림으로 표현된 정보를 이용할 수 없기 때문에 인접 행렬이나 인접 리스트에 의해 표현됨
- 이를 통하여 컴퓨터 프로그램으로 구현됨

### 정의 인접 행렬

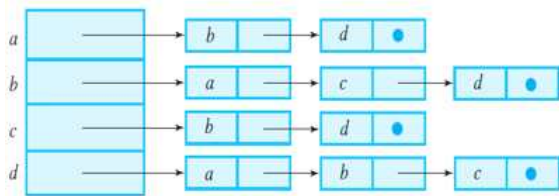
$G = (V, E)$ : 그래프,  $|V| = n$

$\Rightarrow A = (a_{ij})_n$ : 인접행렬

$$a_{ij} = \begin{cases} 1, & (v_i, v_j) \in E \\ 0, & \text{그외에} \end{cases}$$

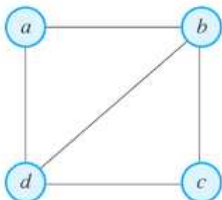
### 정의 인접 리스트

- 각 정점에 대해 포인터가 주어지고, 그 점으로부터 연결된 정점들을 차례로 연결 리스트(linked list)로 표시함
- 같은 리스트 내에서는 순서에 관계가 없음



### 예제

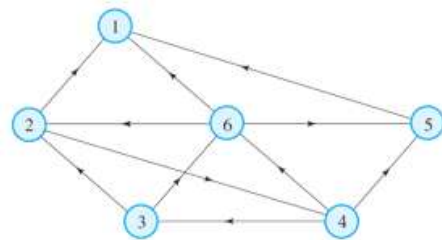
7. 다음과 같은 그래프의 인접 행렬을 구해보자.



8. 다음과 같은 인접 행렬을 가지는 그래프를 그려보자.

$$\begin{matrix} & a & b & c & d & e \\ \begin{matrix} a \\ b \\ c \\ d \\ e \end{matrix} & \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix} \end{matrix}$$

9. 다음의 방향 그래프에서 인접 리스트를 구해보자.



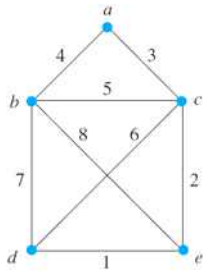
## 7.4 특수 형태의 그래프

교과목명	이산수학	분반		담당교수	김 외 현
학부(과)		학번		성명	

### 정의 오일러 성질을 만족하는 특수한 그래프

#### (1) 오일러 경로

그래프에서 각 연결선을 단 한 번씩만 통과하는 경로



#### (2) 오일러 순회

그래프에서 정점은 여러 번 지날 수 있지만, 각 연결선을 단 한 번씩만 통과하는 순회

### 정리 정점들의 차수의 합

$G = (V, E)$ : 그래프,  $\forall v \in V$ : 정점

$$\Rightarrow \sum \deg(v) = 2 \times (\text{연결선들 개수})$$

### 정리 오일러 경로를 가지기 위한 필요충분조건

$G = (V, E)$ : 오일러 경로

$\Leftrightarrow G$ : 연결 그래프,

$\deg(v)$ =(홀수)인  $v \in V$ 가 0개 또는 2개

### 정리 오일러 순회를 가지기 위한 필요충분조건

$G = (V, E)$ : 오일러 순회

$\Leftrightarrow G$ : 연결 그래프,

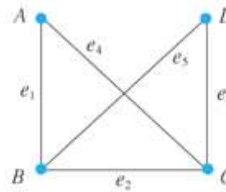
$\forall v \in V, \deg(v)$ =(짝수)

### 정의 한붓그리기

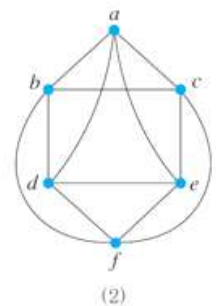
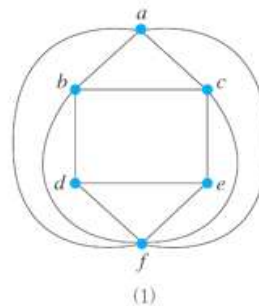
- 그래프에서 연필을 떼지 않고 모든 변을 오직 한 번만 지나는 것
- 연결 그래프에서 한붓그리기가 가능하려면 시작점과 끝점을 제외한 모든 정점의 차수가 짝수

### 예제

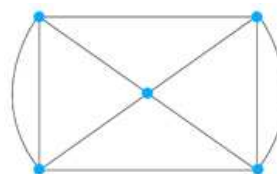
10. 다음 그래프에서 모든 정점들의 차수의 합은 연결선들 개수의 2배임을 보이자.



11. 다음과 같은 두 그래프들이 오일러 경로를 가지는지를 확인해보자.



12. 다음 그래프는 한붓그리기가 가능한지를 판별해보자.

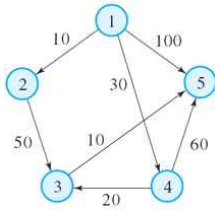


**정의** 해밀턴 성질을 만족하는 특수한 그래프

- (1) 해밀턴 경로  
그래프에서 모든 정점을 오직 한 번씩만 지나지만 시작점으로 돌아오지 않는 경로
- (2) 해밀턴 순회  
그래프에서 모든 정점들을 오직 한 번씩만 지나는 순회

**정의** 가중 그래프

그래프  $G$ 의 연결선에 0보다 큰 수(가중치)를 할당한 그래프



**정의** 동형 그래프, 준동형 그래프

- (1)  $G_1 = (V_1, E_1)$ ,  $G_2 = (V_2, E_2)$ : 그래프  
 $\exists f: V_1 \rightarrow V_2$   
 $(u, v) \in E_1 \Leftrightarrow (f(u), f(v)) \in E_2$  만족하는  
전단사함수  
 $\Rightarrow G_1, G_2$ : 동형 그래프
- (2)  $G', G''$ : 그래프  $G$ 의 연결선 위에 정점을  
추가해 만든 그래프  
 $\Rightarrow G', G''$ :  $G$ 의 준동형 그래프

**정의** 평면 그래프

평면상에서 어떠한 연결선들도 서로 교차할 수 없도록 그려진 하나의 그래프

**정리** 오일러의 정리

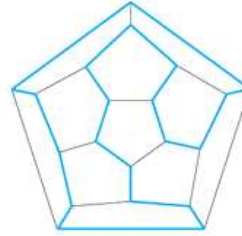
$v$ : 정점의 수  
 $e$ : 연결선의 수  
 $f$ : 면의 수

$$\Rightarrow v - e + f = 2$$

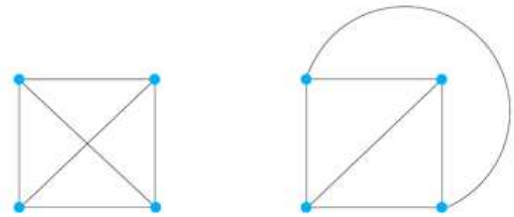
**주의**  $f$ : 그래프 바깥에 있는 면도 포함

**예제**

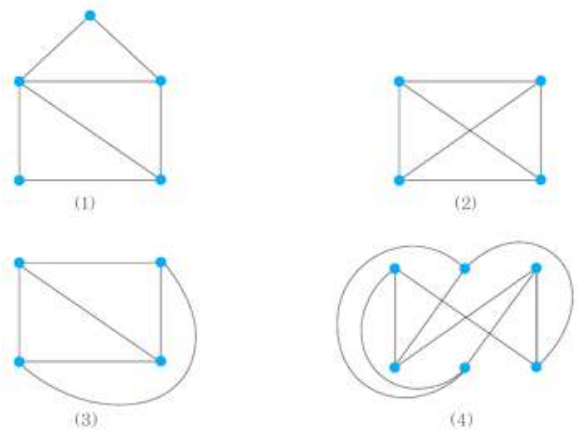
13. 다음 그래프에서 굵은 색선으로 표현된 연결선은 해밀턴 순회를 나타냄을 알아보자.



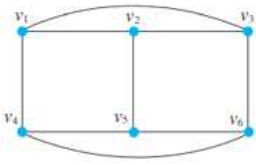
14. 다음 두 그래프가 동형 그래프인지 알아보자.



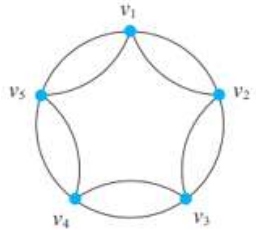
15. 다음 그래프들이 평면 그래프인지의 여부를 알아보자.



16. 그래프에서 오일러의 정리가 성립하는지를 살펴보자.

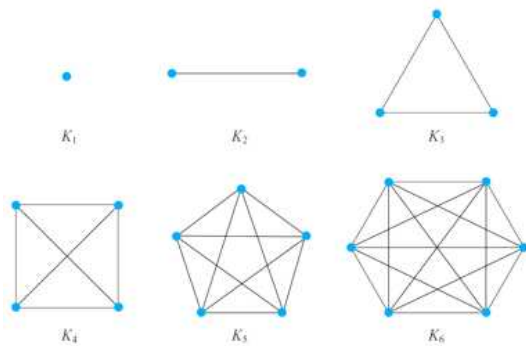


17. 다음 그래프에서 오일러의 공식을 이용하여 오일러의 정리가 성립하는지를 살펴보자.



### 정의 완전 그래프

모든 정점들이 서로 연결되어 있는 그래프



### 참고 완전 그래프에서 연결선의 개수

$K_n$ :  $n$ 개의 정점으로 구성된 완전 그래프

$$\Rightarrow (\text{연결선의 개수}) = \frac{n(n-1)}{2}$$

### 예제

18. 다음과 같은 완전 그래프의 연결선의 개수를 구해보자.



### 정의 정규 그래프

모든 정점의 차수가 같은 그래프

### 참고 $G = (V, E)$ : 그래프,

$$\forall v \in V, \deg(v) = n$$

$\Rightarrow G$ :  $n$ 차 정규 그래프

### 정의 이분 그래프, 완전 이분 그래프

$G = (V, E)$ : 그래프,

$$X \cup Y = V, X \cap Y = \emptyset (X, Y: V \text{의 분할})$$

(1) 이분 그래프

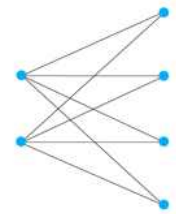
모든 정점이  $X$ 의 한 정점에서  $Y$ 의 어떤 정점으로 연결되는 그래프

(2) 완전 이분 그래프

$X$ 의 모든 정점과  $Y$ 의 모든 정점 사이에 연결선이 존재하는 그래프



(1)  $K_{2,3}$

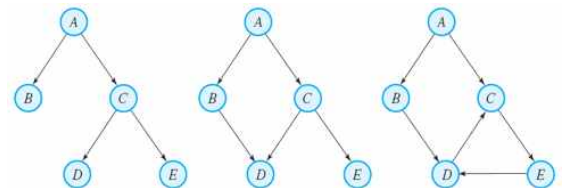


(2)  $K_{2,4}$

### 정리 방향 비사이클 그래프(DAG)

사이클이 없는 방향 그래프

트리보다는 일반적이거나 임의의 방향 그래프보다는 제한적



트리

DAG

방향 그래프

### 예제

19. 집합  $\{1, 2, 3\}$ 의 진부분 집합을 DAG로 표현해보자.



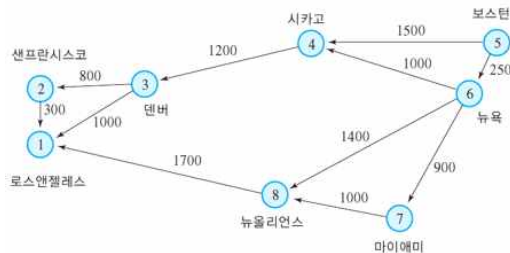
## 7.5 그래프의 응용

교과목명	이산수학	분반		담당교수	김 의 현
학부(과)		학번		성명	

### 참고 그래프의 응용 예

#### (1) 최단 경로 문제

- 그래프는 그림과 같이 지도에서 도시를 나타내는 점과 그들 도시 간의 거리를 나타내는 연결선의 값으로 표현됨



- 도시 A에서 도시 B로 가기 위한 방법
- 첫째, A에서 B로 가는 경로가 있느냐는 점
- 둘째, A에서 B로 가는 경로가 여러 개 있을 경우 어떤 경로로 가는 것이 가장 짧은 거리인가 하는 점
- 가장 짧은 거리의 경로를 찾는 문제를 최단 경로 문제라 함
- 주어진 방향 그래프에서 경로의 시작점을 출발점이라 하며 목적지를 도착점이라고 함
- 주어진 연결선의 길이는 0 이상인 경우를 가정함

### 정의 다익스트라 알고리즘

- 최단 경로의 거리 문제를 해결할 수 있는 방법
- 하나의 정점에서 다른 정점까지의 최단 경로를 구하는 방법
- 주어진 방향 그래프  $G = (V, E)$ 에서  $V = \{1, 2, \dots, n\}$ 이고 점  $\{1\}$ 이 출발점이라고 가정함
- 점  $i$ 에서  $j$ 로 가는 거리를  $C[i, j]$ 로 나타내는데, 만약  $i$ 에서  $j$ 로 가는 경로가 없으면 거리는  $\infty$ 가 됨
- $D[i]$ 는 출발점에서 현재점  $i$ 에 이르는 가장 짧은 거리를 나타냄

#### 알고리즘 1 다익스트라 알고리즘

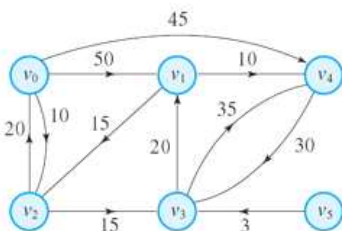
```

void Dijkstra()
/* 이 알고리즘은 정점 1에서 방향 그래프의 모든 정점까지의 최단 거리를 계산한다. */
{
    S = {1};
    for( i = 2 ; i <= n ; i++ )
        D[i] = C[1, i]; /* D값을 초기화한다 */
    for( i = 1 ; i <= n-1 ; i++ )
    {
        choose a vertex w in V - S such that
            D[w] is a minimum;
        add w to S;
        for(each vertex v in V - S)
            D[v] = min (D[v], D[w] + C[w, v]);
    }
} /* Dijkstra */

```

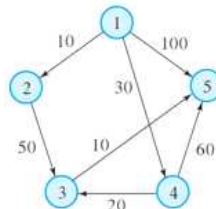
### 예제

20. 주어진 그래프에서 연결선의 값이 두 점 사이의 거리를 나타낼 때,  $v_0$ 가 출발점이라 하면  $v_0$ 에서 나머지 각 정점까지의 최단 경로와 거리를 구해보자.



### 예제

21. 다익스트라 알고리즘을 다음과 같은 가중 그래프에 적용해보자.



## 참고 그래프의 응용 예

### (2) 해밀턴 순회의 응용

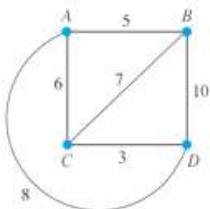
- 해밀턴 순회의 응용 문제로는 순회판매원 문제가 있음
- 순회판매원 문제란 방문해야 할 도시들과 이들 사이의 거리가 주어졌을 경우, 순회판매원이 어떤 특정한 도시를 출발하여 어떠한 도시도 두 번 방문함이 없이 모든 도시들을 거쳐 처음 출발한 도시로 되돌아올 때, 총 여행 거리가 최소가 되는 경로를 찾는 문제임
- 최소의 경로가 최적의 경로라고 할 수 있음
- 일반적인 해결 알고리즘이 존재하지 않음
- 최근접 이웃 방법을 통하여 최솟값은 아니더라도 근사값은 구할 수 있음
- 임의로 선택한 꼭짓점에서 출발하여 그 꼭짓점과 가장 가까운 꼭짓점을 찾아서 연결하고 그 경로를 첨가하는 과정을 반복하며 마지막에 순회를 형성하도록 하는 것임

### 알고리즘 2 최근접 이웃 방법 알고리즘

```
begin
  Choose any  $v_1 \in V$ .
   $v' \leftarrow v_1$ 
   $w \leftarrow 0$ 
  Add  $v'$  to the list of vertices in the path.
  while unmarked vertices are remained do
    begin
      Mark  $v'$ .
      Choose any unmarked vertex,  $u$ , that is closest to  $v'$ .
      Add  $u$  to the list of vertices in the path.
       $w \leftarrow w + \text{the weight of the edge } (v', u)$ 
       $v' \leftarrow u$ 
    end
  Add  $v_1$  to the list of vertices in the path.
   $w \leftarrow w + \text{the weight of the edge } (v', v_1)$ 
end.
```

## 예제

22. 다음의 가중 그래프에서 최근접 이웃 방법을 적용한 해밀턴 순회를 만들어보자.



## 7.6 그래프의 탐색

교과목명	이산수학	분반		담당교수	김 의 현
학부(과)		학번		성명	

### 정의 그래프의 각 정점을 한 번씩만 방문하는 방법

#### (1) 깊이 우선 탐색 (DFS)

- 깊이 우선 탐색은 시작점  $v$ 부터 방문함
- $v$ 에 인접한 정점 중에서 방문하지 않은 정점  $w$ 를 방문하고 다시  $w$ 로부터 탐색을 시작함
- 그 후 어떤 정점  $u$ 를 방문하고  $u$ 에 인접한 모든 정점들을 이미 방문한 경우에는 그 전에 마지막으로 방문한 정점으로 되돌아가서 위의 과정들을 반복함
- 모든 정점들을 방문한 후 탐색을 종료함

### 참고 DFS 알고리즘

- 순차적인 프로그램보다는 재귀 알고리즘으로 구현하는 것이 좋음
- 재귀 알고리즘으로 구현할 경우에는 스택을 사용함

#### 알고리즘 3 DFS

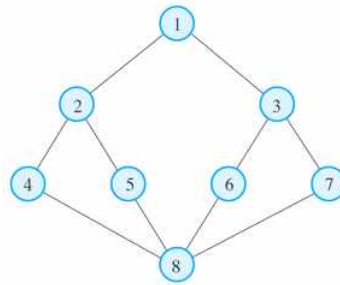
```
void dfs(int v)
/* G = (V, E)가 n개의 정점을 가진 그래프이고 처음에는 False값으로 행렬
visited[n]이 주어졌다고 할 때, 이 알고리즘은 정점 v로부터 도달 가능한 모든 정
점들을 방문한다. */
{
    int w;
    visited[v] = TRUE;
    for(each vertex w adjacent to v)
        if(!visited[w]) dfs(w);
} /* dfs */
```

#### (2) 너비 우선 탐색 (BFS)

- 처음에 방문한 정점과 인접한 정점들을 차례로 방문함 (DFS와의 차이점)
- 먼저 시작점  $v$ 를 방문한 후  $v$ 에 인접한 모든 정점들을 차례로 방문함
- 더 이상 방문할 정점이 없는 경우 다시  $v$ 에 인접한 정점 가운데 맨 처음으로 방문한 정점과 인접한 정점들을 차례로 방문함
- 다음으로  $v$ 에 인접한 정점 중 두 번째로 방문한 정점과 인접한 정점들을 차례로 방문하는 과정을 반복함
- 모든 정점들을 방문한 후 탐색을 종료함

### 예제

23. 다음 그래프를 정점 1에서 시작하여 깊이 우선 탐색과 너비 우선 탐색으로 탐색해보자.



## 7.7 그래프의 색칠 문제

교과목명	이산수학	분반		담당교수	김 의 현
학부(과)		학번		성명	

### 정의 그래프의 색칠 문제

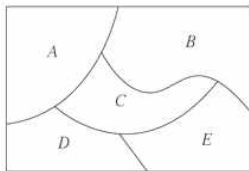
그래프  $G$ 의 인접한 어느 두 영역도 같은 색이 안 되도록 각 정점에 색을 칠하는 문제

### 정의 그래프의 색칠 수

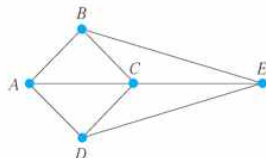
$\chi(G) = (G \text{를 색칠하는데 필요한 색의 최소수})$

### 참고 지도를 색칠하는 문제

- 지도를 색칠하는 문제는 그래프의 색칠 문제와 연관시킬 수 있음
- 예를 들어, (1)의 그래프와 동형이 되는 (2)의 그래프를 쌍대 그래프라고 함



(1)



(2)

- 지도의 영역을 색칠하는 문제는 쌍대 그래프의 정점들을 색칠하는 문제와 동일하므로 그래프의 인접한 어떤 정점들도 같은 색깔을 가지지 않도록 해야 함

### 참고 웰치-포웰 알고리즘

- (1) 그래프의 정점의 차수가 내림차순이 되게 배열
- (2) 배열의 첫 번째 정점은 첫 번째 색으로 색칠하고 계속해서 배열의 순서대로 이미 색칠된 정점과 인접하지 않은 정점을 모두 같은 색으로 색칠
- (3) 배열에서 먼저 나타나는 색칠되지 않은 정점을 두 번째 색으로 색칠하고, 계속해서 배열의 순서대로 지금 색칠하고 있는 색으로 이미 색칠된 정점과 인접하지 않은 정점을 모두 색칠
- (4) 그래프의 모든 정점이 색칠될 때까지 반복

### 정리 2색 가능 동치조건

$G$ : 그래프

(1)  $G$ : 2색 가능

$\Leftrightarrow$  (2)  $G$ : 이분 그래프

$\Leftrightarrow$  (3)  $G$ : 모든 순환의 길이는 짝수

### 정리 4색 정리

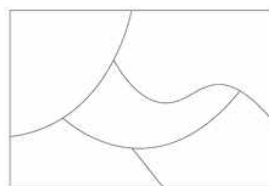
$\forall G$ : 평면 그래프  $\Rightarrow G$ : 4색 가능

### 참고 그래프 색칠하기의 3가지 응용 분야

- (1) 화학물질을 창고에 저장할 때 만일의 사고를 방지하기 위해 서로 반응하는 물질끼리는 다른 장소에 배치하는 문제
- (2) 방송국들의 주파수가 같아서 난시청 지역이 발생하지 않도록 하는 문제
- (3) 애완동물을 판매할 때 사이가 좋지 않은 동물들을 다른 전시관에 배치하는 문제

### 예제

24. 다음 그림과 같이 경계가 주어진 지도를 색칠해보자. 서로 경계가 만나는 부분이 있으면 서로 다른 색을 사용해야 한다.



25. 웰치-포웰 알고리즘을 이용해 다음 그래프를 색칠해보자.

