

CHAPTER 09 트리

학습개요

- 기본 개념

- ◆ 그래프의 특수한 형태인 트리에서 사용되는 용어와 개념을 파악한다

- 이진트리

- ◆ 이진트리의 여러 가지 종류와 순회 방법을 살펴본다
- ◆ 원소들을 특별한 순서로 구성한 이진탐색트리를 이해한다
- ◆ 이진트리로 정의되는 허프만 코드를 이해하고 허프만 트리를 만든다

- 신장트리

- ◆ 깊이우선탐색과 너비우선탐색 알고리즘을 이용하여 신장트리를 구한다

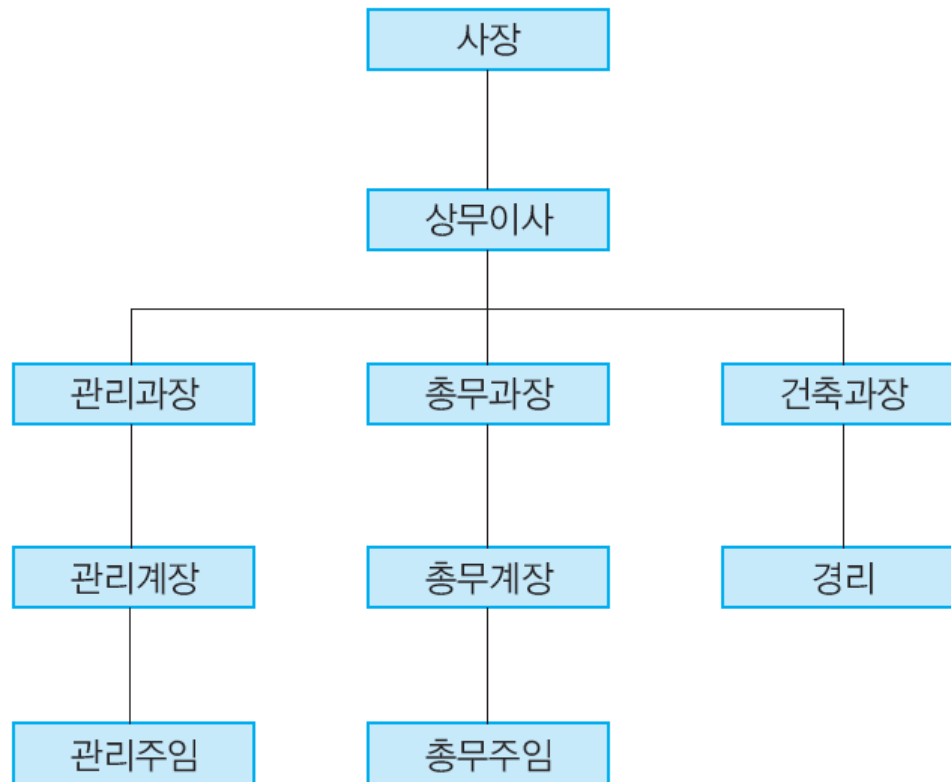
- 최소신장트리

- ◆ 프림 알고리즘과 크루스칼 알고리즘을 이용하여 최소신장트리를 구한다

Section 01 기본 개념 (1)

● 트리(tree)

- ◆ 가계도나 회사의 조직도 등을 나타낼 때 유용하게 사용되고 있는 비선형 자료구조(non-linear data structure)
- ◆ 계층적 자료구조(hierarchical data structure)

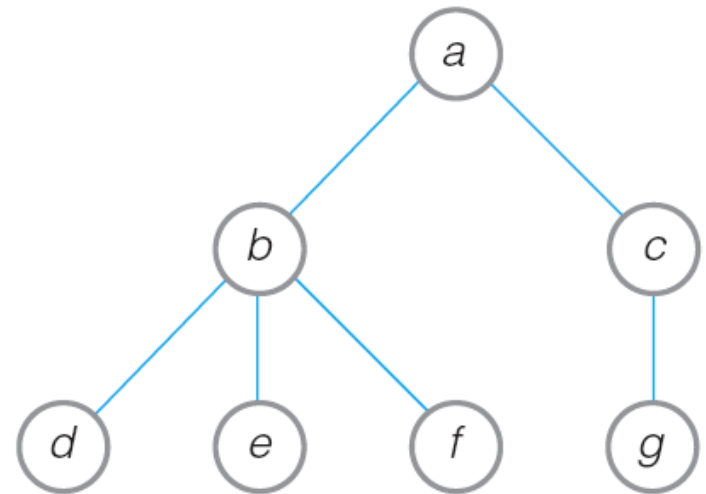


Section 01 기본 개념 (2)

정의 9.1

트리(tree)는 루트(root)라는 특별한 노드(node)가 하나 있는 비순환(acyclic), 연결.connected) 그래프다.

- 노드(node)
 - ◆ 트리를 구성하는 정점(vertex)
- 루트(root)
 - ◆ 트리의 가장 상위에 있는 노드

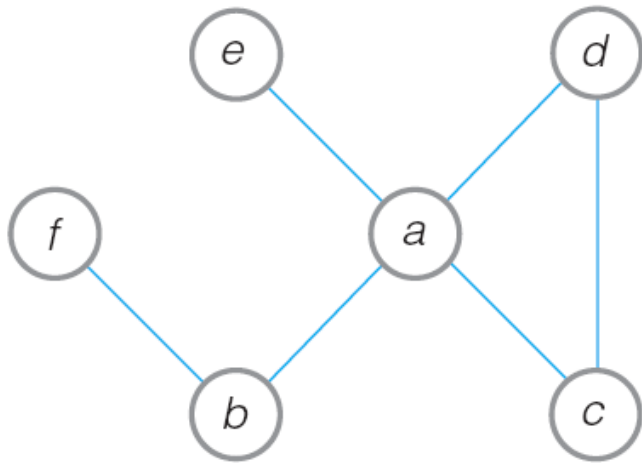


Section 01 기본 개념 (3)

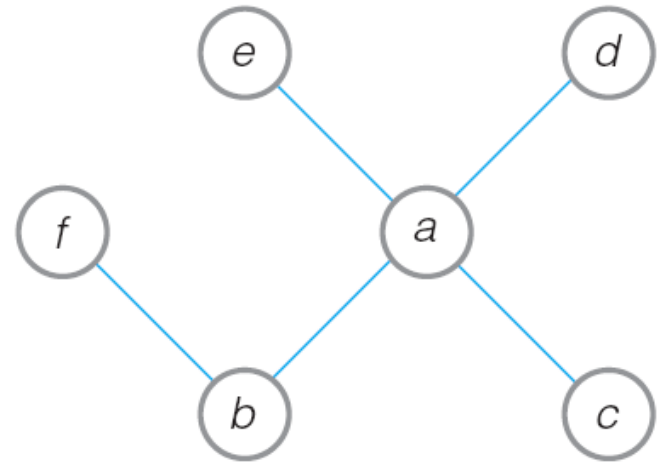
예제 9.1

다음 그래프들 중에서 트리를 찾아라.

(1)

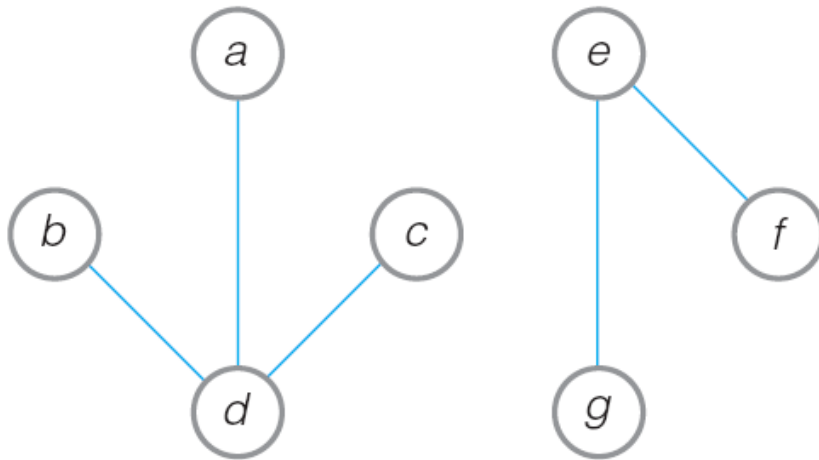


(2)



Section 01 기본 개념 (4)

(3)



풀이

(1) 순환 (a, c, d) 를 가지므로 트리가 아니다.

(2) 비순환인 연결그래프므로 트리다.

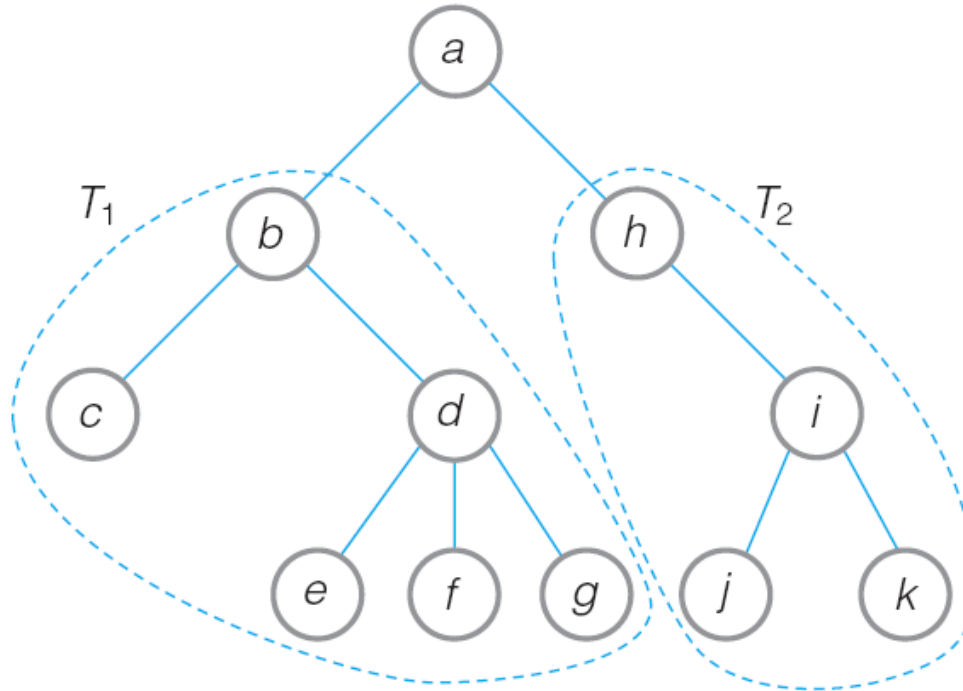
(3) 연결되어 있지 않으므로 트리가 아니다.

● 숲(forest)

- ◆ 연결되어 있지 않아서 트리는 아니지만 그래프의 각 성분은 트리되어 있는 그래프

Section 01 기본 개념 (5)

- 서브트리(subtree)



- 노드차수(node degree)

- ◆ 하나의 노드가 가지고 있는 서브트리의 수

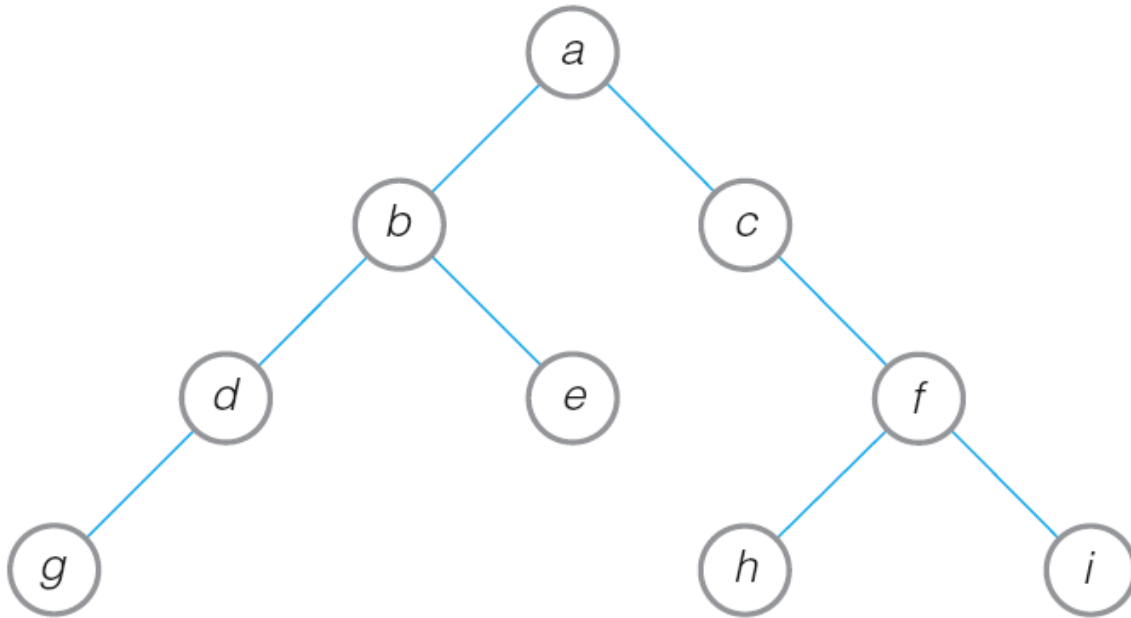
Section 01 기본 개념 (6)

- 트리의 차수(degree of tree)
 - ◆ 트리에 있는 노드차수 중에서 가장 큰 차수
- 리프노드(leaf node) 또는 단말노드(terminal node)
 - ◆ 트리에서 자식노드를 가지고 있지 않은 노드, 즉 차수가 0인 노드
- 트리의 깊이(depth) 또는 높이(height)
 - ◆ 트리에서 가장 큰 레벨(level)
- 조상(ancestor)
 - ◆ 해당 노드에서 루트에 이르는 경로 위에 있는 모든 노드
- 자손(descendant)
 - ◆ 해당 노드와 연결되어 있는 하위 레벨의 모든 노드

Section 02 이진트리 (1)

정의 9.2

모든 노드의 차수가 n 이하인 트리를 n 항트리(n -ary tree)라고 하고, $n=2$ 일 때 이진트리(binary tree)라고 한다.

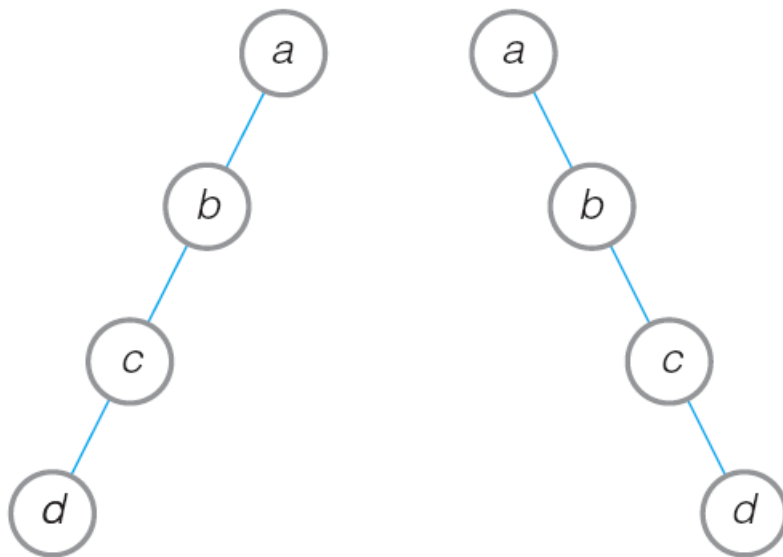


Section 02 이진트리 (2)

정의 9.3

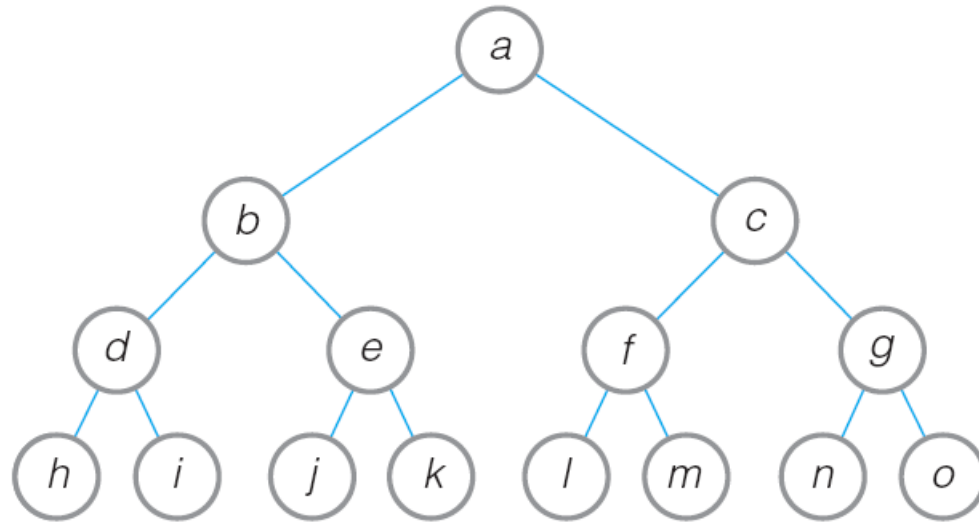
모든 노드가 왼쪽 자식만을 갖거나 오른쪽 자식만을 가지는 이진트리를 사향이진트리(skewed binary tree)라고 하며, 트리의 깊이가 n 인 이진트리의 전체 노드 수가 그 트리의 최대 노드 수 $2^{n+1}-1$ 과 같으면 포화이진트리(full binary tree)라고 한다. 또한 트리의 깊이가 n 인 이진트리의 전체 노드 수가 그 트리의 최대 노드 수 $2^{n+1}-1$ 보다 작으며, 마지막 레벨 이외의 모든 레벨은 노드가 모두 차 있고 마지막 레벨에서는 왼쪽부터 노드가 차례대로 차 있으면 완전이진트리(complete binary tree)라고 한다.

● 사향이진트리

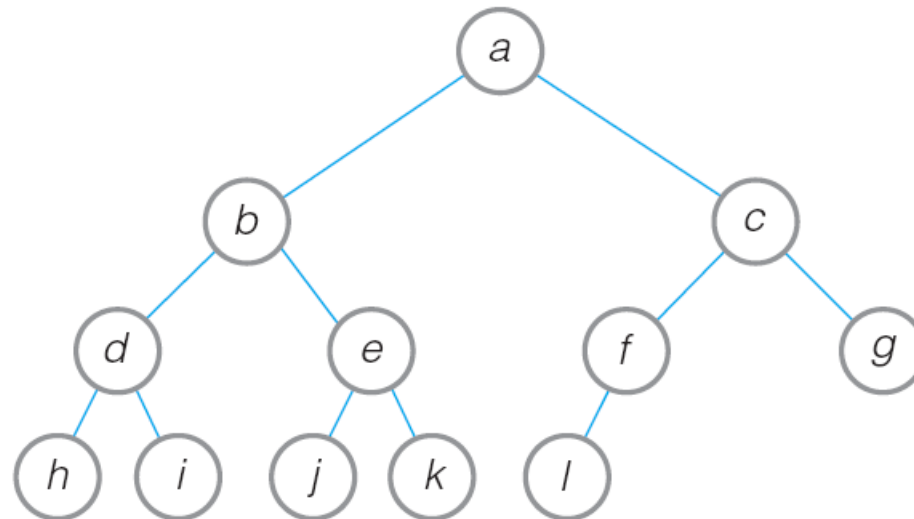


Section 02 이진트리 (3)

- 포화이진트리

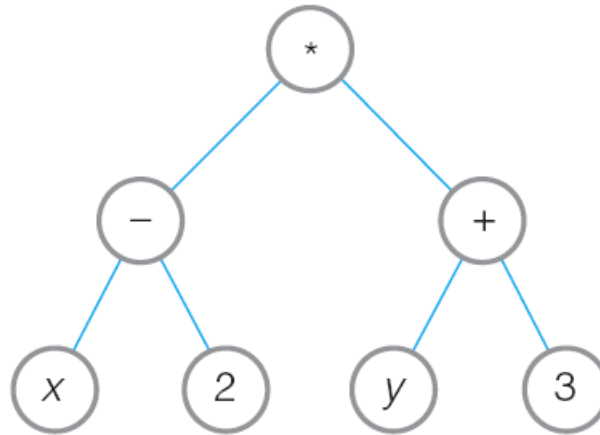


- 완전이진트리



Section 02 이진트리 (4)

- $(x-2)*(y+3)$ 에 대한 이진트리

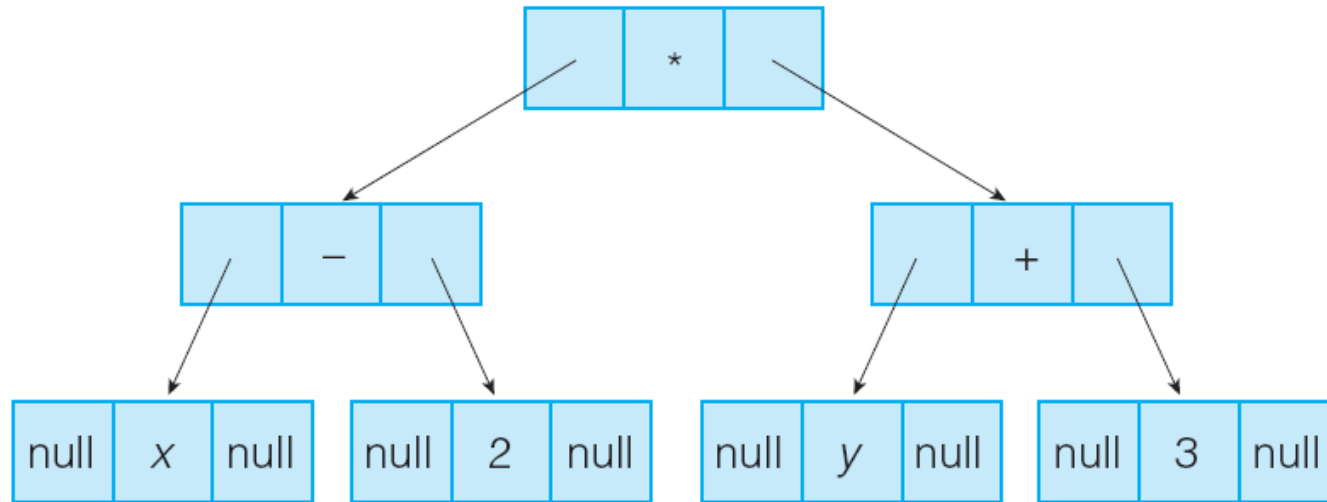


- 이진트리의 배열 표현

	왼쪽 자식	오른쪽 자식
*	-	+
-	x	2
+	y	3
x	0	0
2	0	0
y	0	0
3	0	0

Section 02 이진트리 (5)

- 이진트리의 연결리스트 표현



Section 02 이진트리 (6)

정의 9.4

전위순회(preorder traversal)는 트리의 루트를 우선 방문한 후에 각 서브트리들을 전위순회를 이용하여 가장 왼쪽 서브트리부터 가장 오른쪽 서브트리로 차례대로 이동하면서 방문하는 방법이다. 다시 말하면 다음과 같다.

- (1) 루트 노드를 방문한다.
- (2) 왼쪽 서브트리를 전위순회한다.
- (3) 오른쪽 서브트리를 전위순회한다.

Section 02 이진트리 (7)

정의 9.5

중위순회(inorder traversal)는 루트의 가장 왼쪽 서브트리를 중위순회로 우선 방문한 후에 루트를 방문하고, 나머지 각 서브트리들을 중위순회를 이용하여 가장 왼쪽 서브트리부터 가장 오른쪽 서브트리로 이동하면서 방문하는 방법이다. 다시 말하면 다음과 같다.

- (1) 왼쪽 서브트리를 중위순회한다.
- (2) 루트 노드를 방문한다.
- (3) 오른쪽 서브트리를 중위순회한다.

Section 02 이진트리 (8)

정의 9.6

후위순회(postorder traversal)는 루트의 각 서브트리들을 후위순회를 이용하여 가장 왼쪽 서브트리부터 가장 오른쪽 서브트리로 차례대로 이동하면서 방문한 후에 루트를 맨 마지막에 방문하는 방법이다. 다시 말하면 다음과 같다.

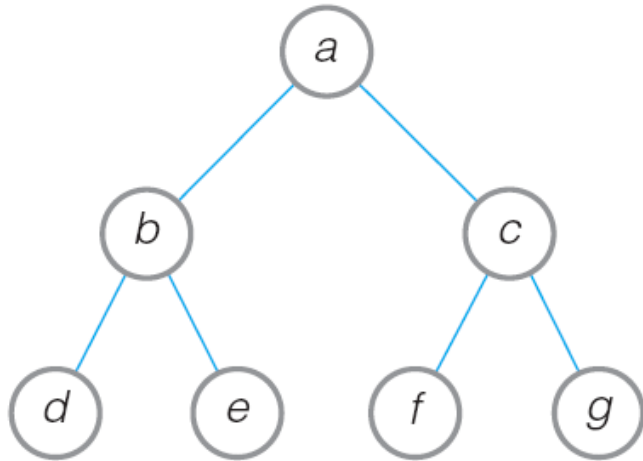
- (1) 왼쪽 서브트리를 후위순회한다.
- (2) 오른쪽 서브트리를 후위순회한다.
- (3) 루트 노드를 방문한다.

Section 02 이진트리 (9)

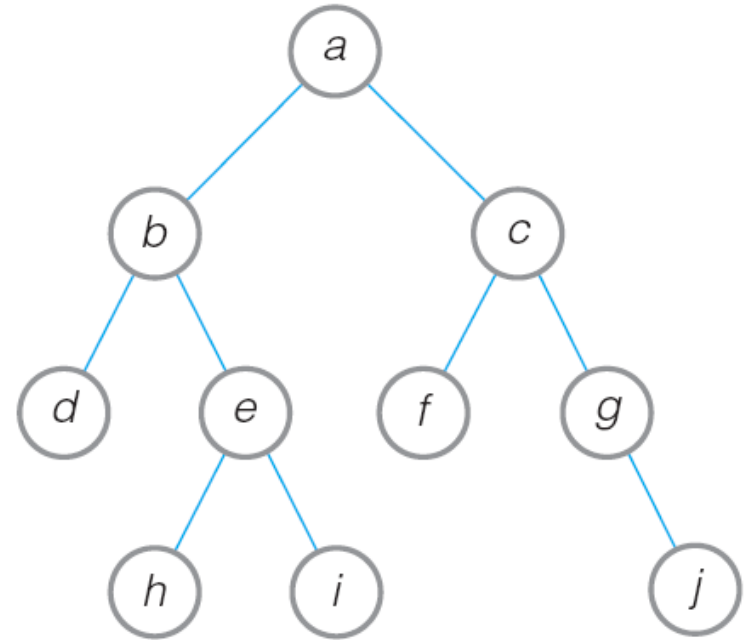
예제 9.5

다음 각 이진트리들에 대하여 전위순회, 중위순회, 후위순회한 결과를 구하여라.

(1)



(2)



Section 02 이진트리 (10)

풀이

(1) 전위순회: $a \rightarrow b \rightarrow d \rightarrow e \rightarrow c \rightarrow f \rightarrow g$

중위순회: $d \rightarrow b \rightarrow e \rightarrow a \rightarrow f \rightarrow c \rightarrow g$

후위순회: $d \rightarrow e \rightarrow b \rightarrow f \rightarrow g \rightarrow c \rightarrow a$

(2) 전위순회: $a \rightarrow b \rightarrow d \rightarrow e \rightarrow h \rightarrow i \rightarrow c \rightarrow f \rightarrow g \rightarrow j$

중위순회: $d \rightarrow b \rightarrow h \rightarrow e \rightarrow i \rightarrow a \rightarrow f \rightarrow c \rightarrow g \rightarrow j$

후위순회: $d \rightarrow h \rightarrow i \rightarrow e \rightarrow b \rightarrow f \rightarrow j \rightarrow g \rightarrow c \rightarrow a$

● 표기법의 형식과 예

표기법	일반적인 형식	예
전위표기법	$\langle \text{operator} \rangle \langle \text{operand} \rangle \langle \text{operand} \rangle$	$+xy$
중위표기법	$\langle \text{operand} \rangle \langle \text{operator} \rangle \langle \text{operand} \rangle$	$x+y$
후위표기법	$\langle \text{operand} \rangle \langle \text{operand} \rangle \langle \text{operator} \rangle$	$xy+$

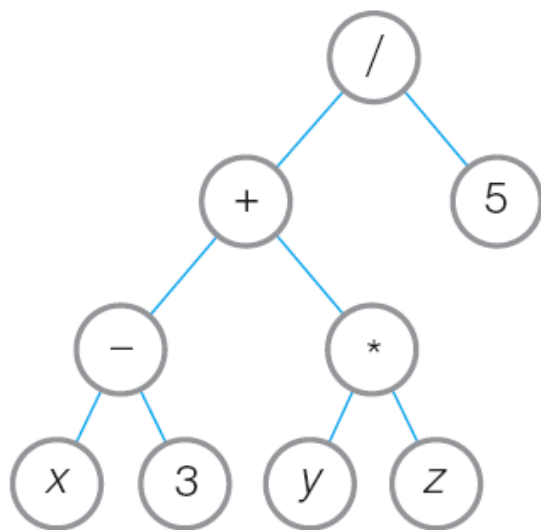
Section 02 이진트리 (11)

예제 9.6

다음의 수식을 이진트리로 나타내고, 전위표기법과 후위표기법으로 나타내어라.

$$((x-3)+(y*z))/5$$

풀이



전위표기법으로 나타내면 다음과 같다.

$$/+ - x 3 * y z 5$$

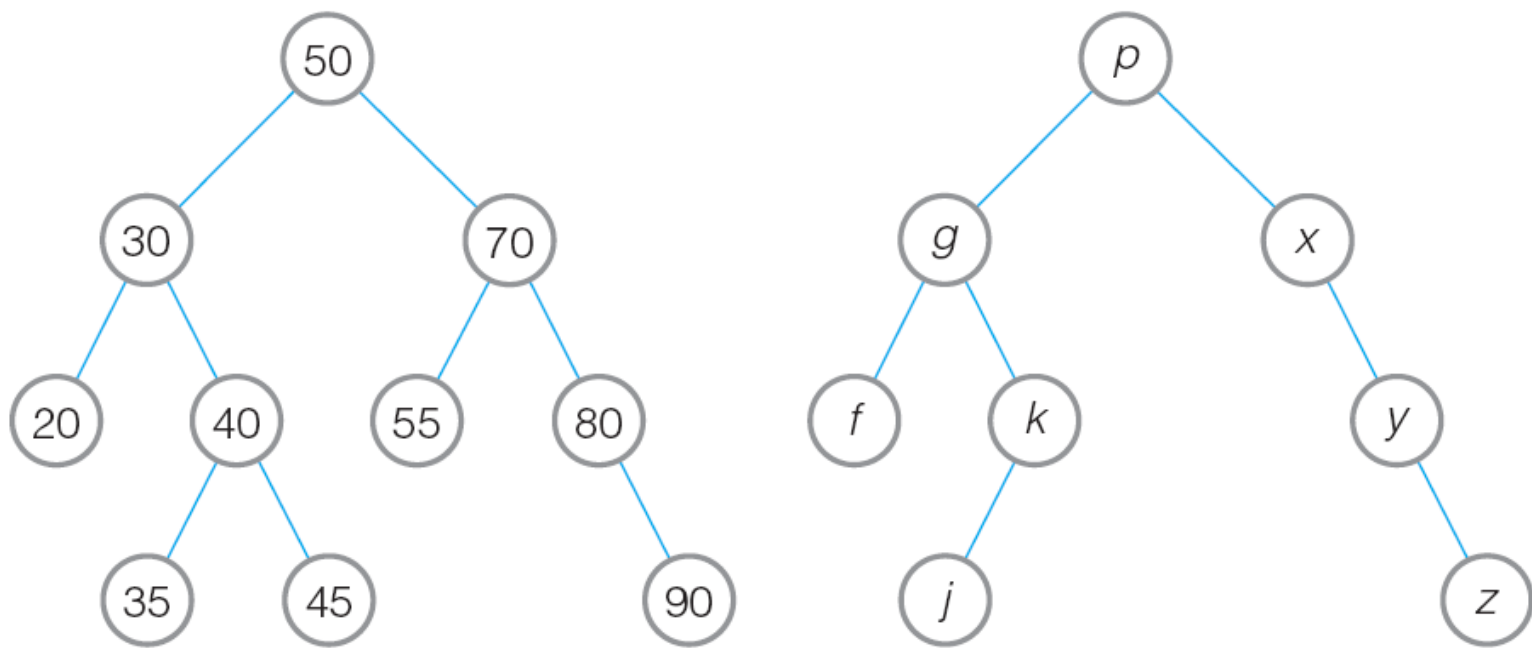
후위표기법으로 나타내면 다음과 같다.

$$x 3 - y z * + 5 /$$

Section 02 이진트리 (12)

정의 9.8

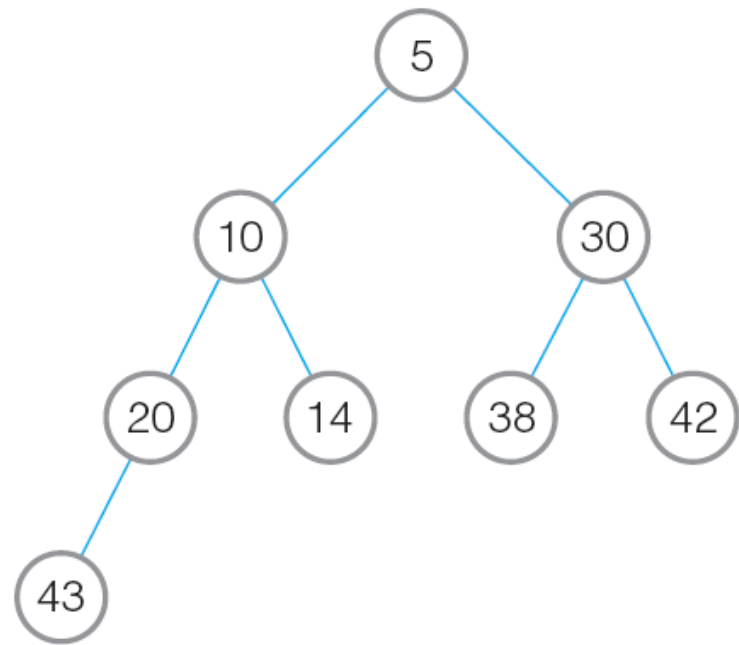
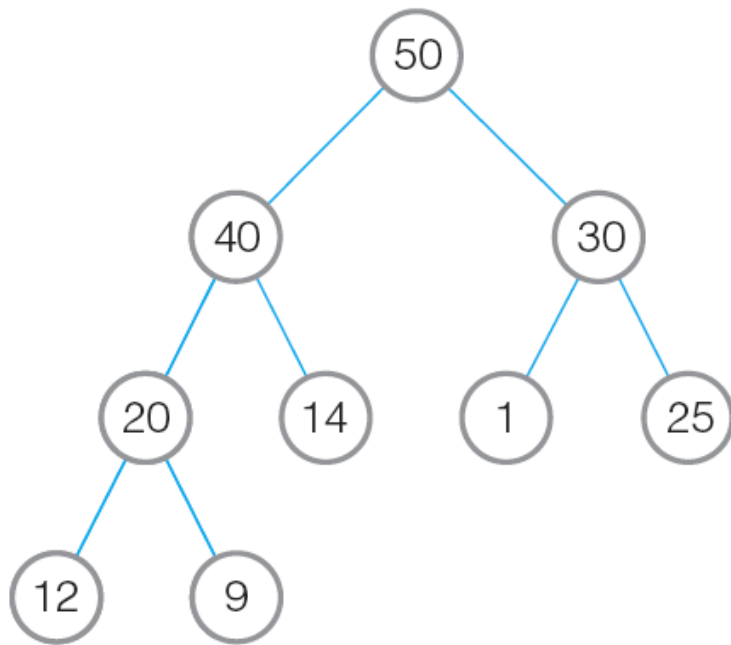
모든 노드 x 에 대하여 노드 y 가 노드 x 의 왼쪽 서브트리에 있을 때는 $y < x$ 고, 노드 z 가 노드 x 의 오른쪽 서브트리에 있을 때는 $x < z$ 인 이진트리를 이진탐색트리(binary search tree)라고 한다. 이때 ' $<$ '는 순서 관계를 의미한다.



Section 02 이진트리 (13)

정의 9.9

각 노드의 값이 자식노드의 값보다 큰 완전이진트리를 최대힙(max heap)이라고 하고, 각 노드의 값이 자식노드의 값보다 작은 완전이진트리를 최소힙(min heap)이라고 한다.



Section 02 이진트리 (14)

● 1차원 배열로 나타낸 최대힙과 최소힙

1	2	3	4	5	6	7	8	9
50	40	30	20	14	1	25	12	9

1	2	3	4	5	6	7	8
5	10	30	20	14	38	42	43

◆ n 개의 노드로 이루어진 힙을 1차원 배열로 나타내고 자신이 저장된 위치의 인덱스를 i 라고 할 때

① $i \neq 1$ 일 때 부모의 위치는 $\lfloor \frac{i}{2} \rfloor$ 다.

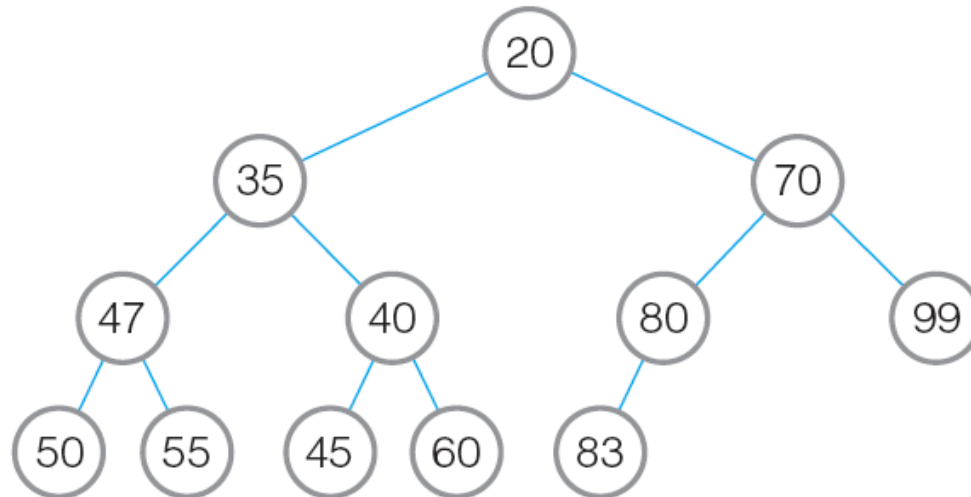
② $2i \leq n$ 일 때 왼쪽 자식의 위치는 $2i$ 다.

③ $2i+1 \leq n$ 일 때 오른쪽 자식의 위치는 $2i+1$ 이다.

Section 02 이진트리 (15)

예제 9.10

다음과 같은 최소힙을 1차원 배열로 나타내고, 인덱스 5에 해당하는 값의 부모, 왼쪽 자식, 오른쪽 자식의 위치와 그 값을 각각 구하여라.



Section 02 이진트리 (16)

풀이 최소힙을 1차원 배열로 나타내면 다음과 같다.

1	2	3	4	5	6	7	8	9	10	11	12
20	35	70	47	40	80	99	50	55	45	60	83

인덱스 5에 해당하는 값인 40의 부모의 위치는 $\lfloor \frac{5}{2} \rfloor = 2$ 므로 부모는 인덱스 2에 해당하는 값 35다. 또한 왼쪽 자식의 위치는 $2 \cdot 5 = 10$ 이므로 45가 왼쪽 자식이 되고, 오른쪽 자식의 위치는 $2 \cdot 5 + 1 = 11$ 이므로 60이 오른쪽 자식이 된다.

Section 02 이진트리 (17)

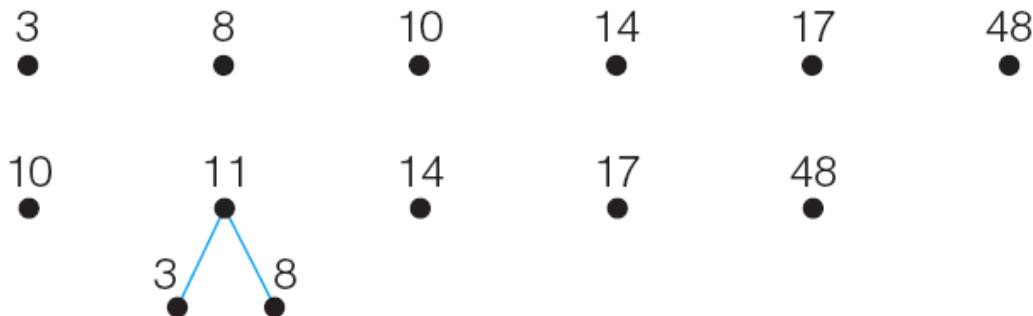
● 데이터 압축(data compression)

- ◆ 발생 빈도가 높은 문자에는 적은 비트를 할당하고, 발생 빈도가 낮은 문자에는 그 보다 많은 비트를 할당하여 저장될 파일의 데이터 크기를 줄이는 것
- ◆ 허프만 코드(Huffman code)가 기본적인 방식

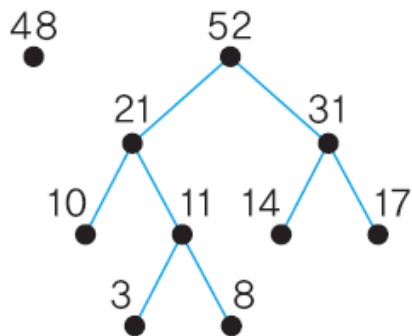
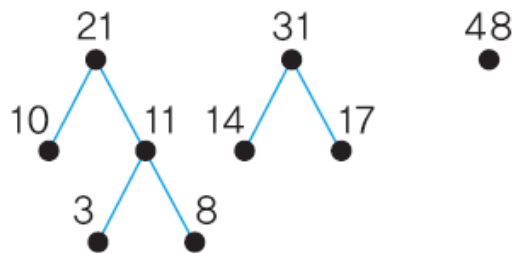
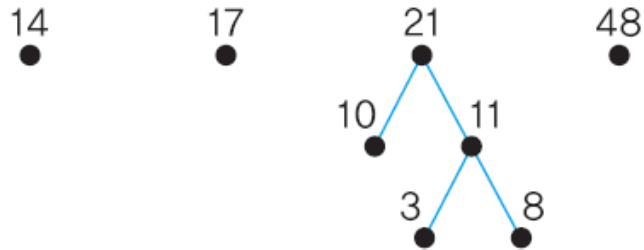
예제 9.13

u, v, w, x, y, z 에 대한 빈도수가 각각 17, 3, 8, 48, 10, 14일 때 이에 대한 허프만 트리를 만들고, 각 문자들의 허프만 코드를 구하여라.

풀이

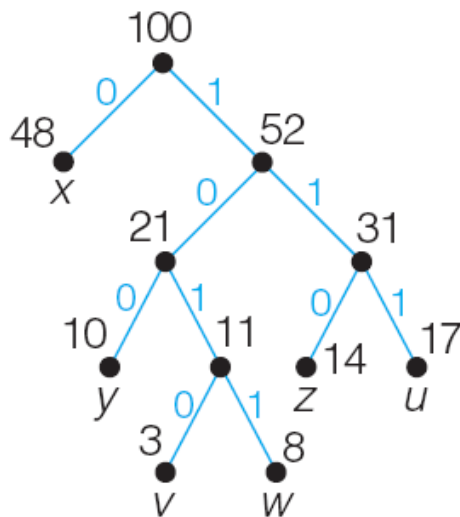


Section 02 이진트리 (18)



Section 02 이진트리 (19)

허프만 트리는 다음과 같다.



허프만 트리에서 u, v, w, x, y, z 에 대한 허프만 코드는 각각 111, 1010, 1011, 0, 100, 110이다.

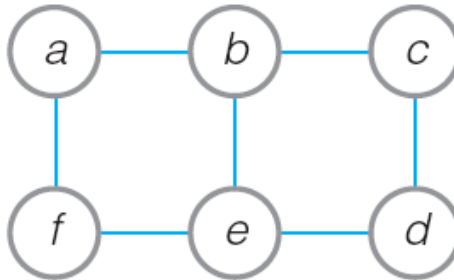
Section 03 신장트리 (1)

정의 9.10

그래프 G 의 모든 정점(노드)들을 포함하는 트리를 신장트리(spanning tree)라고 한다.

예제 9.14

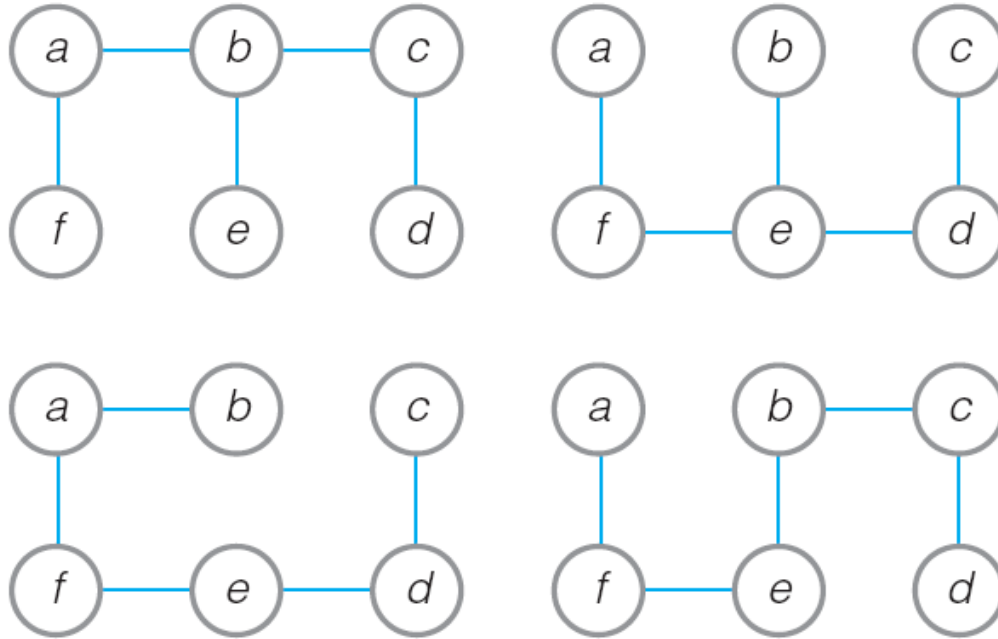
다음 그래프의 신장트리를 그려라.



Section 03 신장트리 (2)

풀이

신장트리는 다음과 같은 여러 가지 형태로 그릴 수 있다.



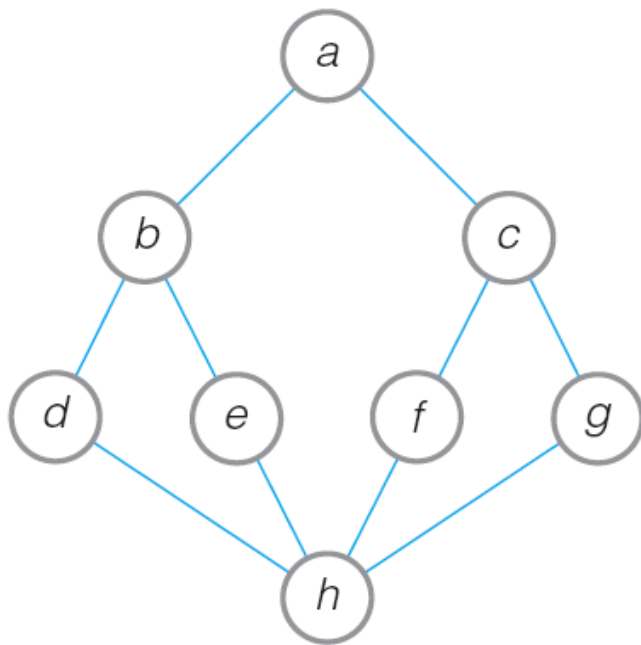
Section 03 신장트리 (3)

- 깊이우선 신장트리(depth first spanning tree)
 - ◆ 깊이우선탐색 알고리즘을 이용하여 만들어진 신장트리
- 너비우선 신장트리(breadth first spanning tree)
 - ◆ 너비우선탐색 알고리즘을 이용하여 만들어진 신장트리

Section 03 신장트리 (4)

예제 9.17

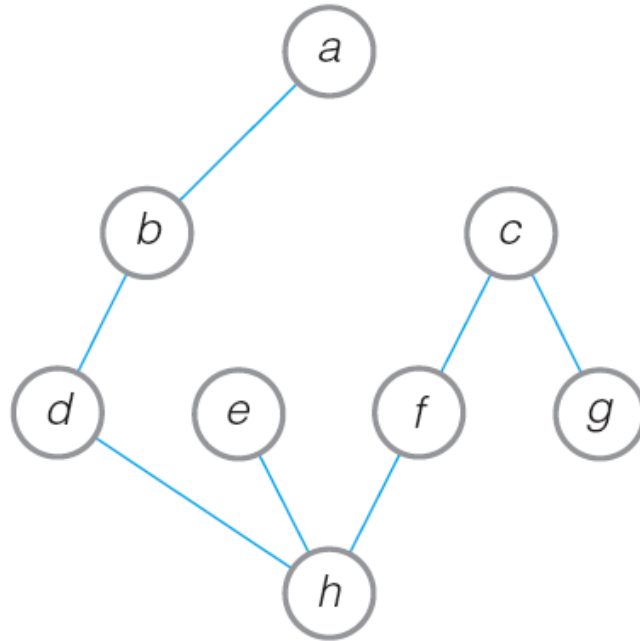
다음 그래프에 대하여 정점 a 를 루트로 하는 깊이우선 신장트리와 너비우선 신장트리를 구하여라.



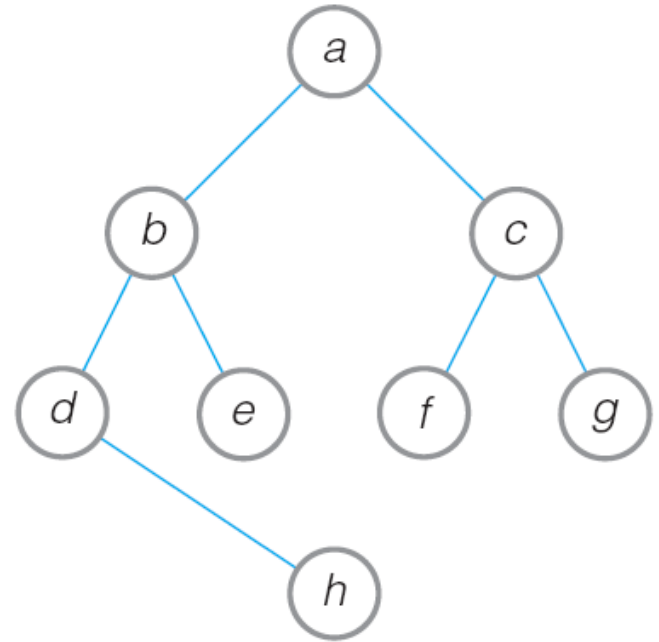
Section 03 신장트리 (5)

풀이

깊이우선 신장트리는 다음과 같다.



너비우선 신장트리는 다음과 같다.

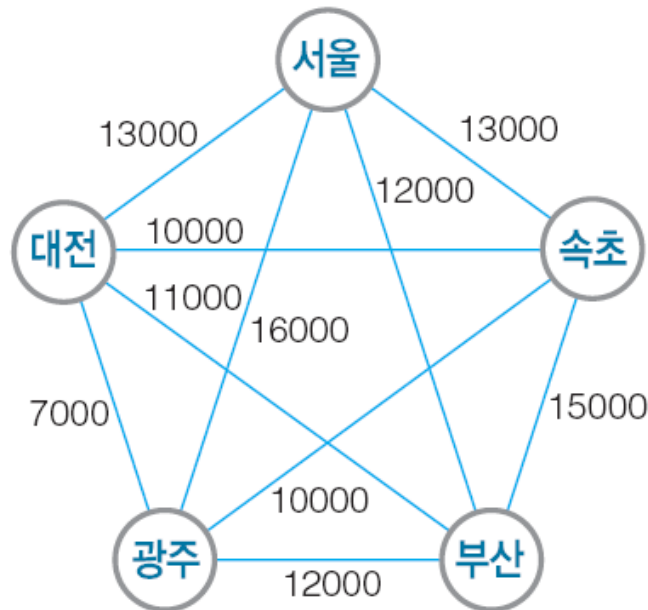


Section 04 최소신장트리 (1)

정의 9.11

신장트리의 비용(cost)은 트리에 포함된 모든 에지의 가중치를 합한 값이며, 최소신장트리(minimal spanning tree)는 비용이 최소가 되는 신장트리다.

- 통신 요금을 가중치로 나타낸 그래프



Section 04 최소신장트리 (2)

● 프림(Prim) 알고리즘

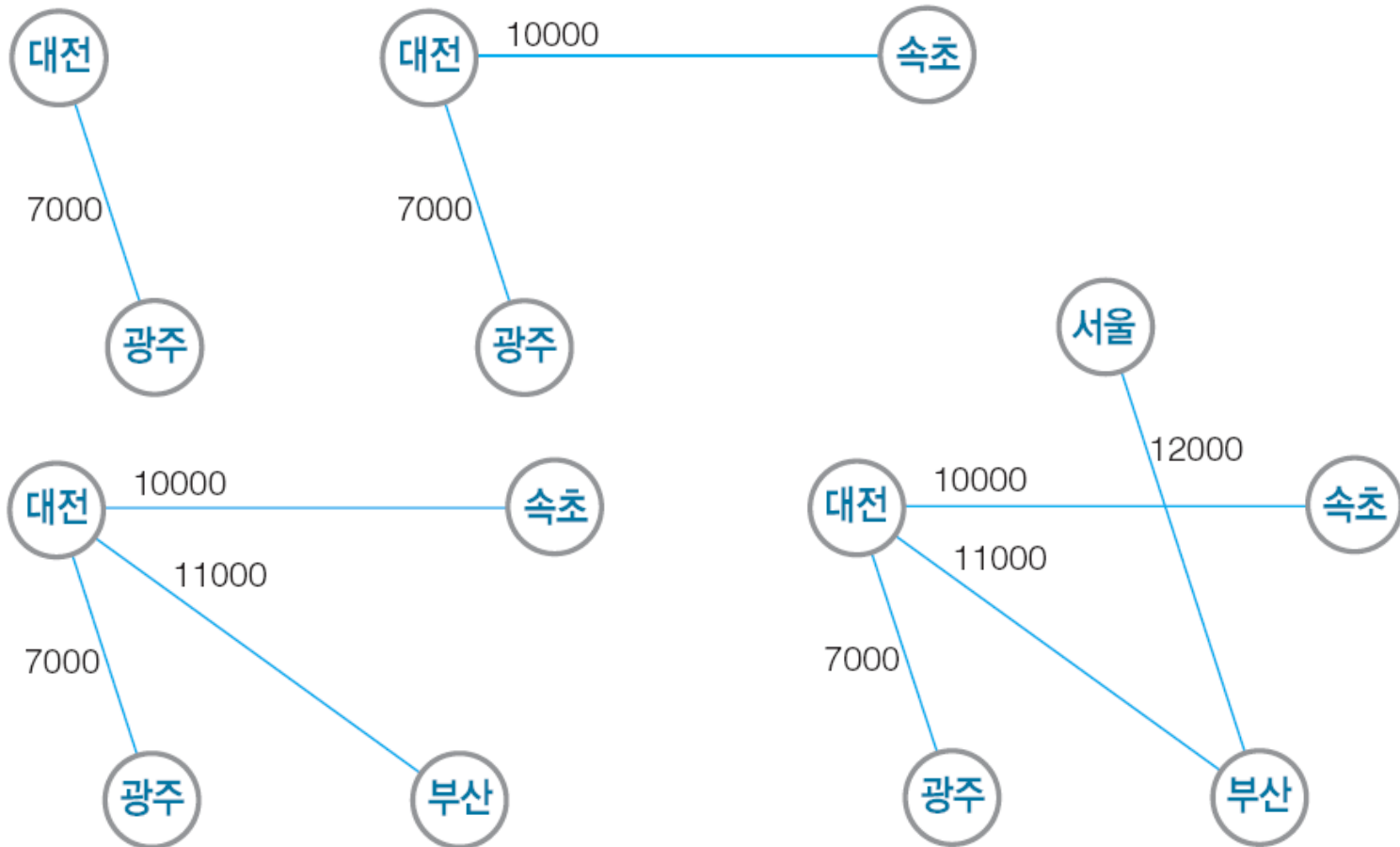
- ◆ 가중치가 가장 작은 에지를 선택하고 이 에지와 연결된 정점에서 다시 가중치가 가장 작은 에지를 선택하여 신장트리에 추가하는 것을 반복함으로써 최소신장트리를 만들어 나간다
- ◆ 트리에 추가되는 에지들은 순환을 형성하지 않아야 한다
- ◆ n 개의 정점에 대하여 $n-1$ 개의 에지가 추가되었을 때 종료된다

Section 04 최소신장트리 (3)

예제 9.18

프림 알고리즘을 이용하여 [그림 9-19]에 대한 최소신장트리를 구하여라.

풀이



Section 04 최소신장트리 (4)

● 크루스칼(Kruskal) 알고리즘

- ◆ 가중치가 가장 작은 에지를 선택하고, 이 에지와 연결되어 있지 않은 에지라도 가중치가 작은 에지를 순서대로 선택하여 신장트리에 추가함으로써 최소신장트리를 만들어나간다
- ◆ 트리에 추가되는 에지들은 순환을 형성하지 않아야 한다
- ◆ n 개의 정점에 대하여 $n-1$ 개의 에지가 추가되었을 때 종료된다

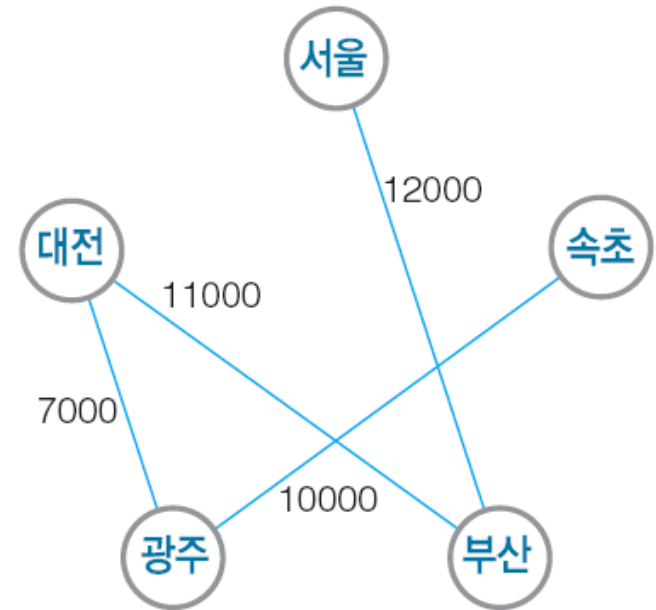
Section 04 최소신장트리 (5)

예제 9.19

크루스칼의 알고리즘을 이용하여 [그림 9-19]에 대한 최소신장트리를 구하여라.

풀이

가중치	에지
7000	{대전, 광주}
10000	{광주, 속초}
10000	{대전, 속초}
11000	{대전, 부산}
12000	{광주, 부산}
12000	{서울, 부산}
13000	{서울, 대전}
13000	{서울, 속초}
15000	{속초, 부산}
16000	{서울, 광주}



Discrete Mathematics

The End

본 강의자료는 강의의 편의를 위해 교수님들께 제공되는 자료입니다. 자료의 글과 그림은 저작권이 저자에게 있으므로 **대중적인 배포를 할 수 없음**을 유의해주시길 바랍니다.

함채원 • 홍영진