정보올림피아드 및 프로그래밍 교육 교재

III 이산수학

제III장에서는 정보올림피아드와 교원프로그래밍경진대회에서 주로 출제되는 이산수학의 기본 개념을 살펴보고 기출문제를 해결한다.

1. 수의 표현

우리들은 일상생활에서 십진법을 사용한다. 십진법은 10개의 숫자 0, 1, 2, ⋯, 9를 사용한다. 예를 들어, 752라는 수가 있다고 하자. 752는 각 자리의 값이 각각 100, 10, 1이므로 수의 자리가 왼쪽으로 하나씩 올라감에 따라 자리의 값이 10배씩 커진다. 이와 같은 수의 표시방법을 십진법이라고 한다.

이진법은 2개의 숫자 0과 1을 사용한다. 이진법은 수의 자리가 왼쪽으로 하나씩 올라감에 따라 자리의 값이 2배씩 커지는 수의 표시방법이다. 0과 1이라고 하는 어느 하나의 값을 갖는 정보의 단위를 비트(bit, binary digit)라고 하고 이러한 정보를 전자 회로에 의해 취급하는 것이 디지털 회로이다.

가. 수의 표현

10의 거듭제곱을 사용하여 십진법의 수를 나타낸 식을 십진법의 전개식이라고 한다. 345.6은 십진법의 전개식으로 나타내면

$$3\times10^2+4\times10^1+5\times10^0+6\times10^{-1}$$

이다. 이진법은 십진법과 구별하기 위해 이진법으로 나타낸 수의 뒷부분에 $_{(2)}$를 표시한다.

예를 들어, $11010_{(2)}$은

$$1\times2^4+1\times2^3+0\times2^2+1\times2^1+0\times2^0$$

으로 나타낸다. 이와 같이 이진법의 수를 2의 거듭제곱을 사용하여 나타낸 식을 이진법의 전개식이라고 한다. 이진법의 전개식을 계산하면 26이 되므로 이진법의 수 $11010_{(2)}$과 십진법의 수 26은 같은 값을 가진다.

일반적으로 n진법에 의한 수의 표현은 다음과 같다. 0,1, 2, ⋯, $n-1$의 수를 사용하고 $a_m a_{m-1} \cdots a_{0\,(n)}$을 n진법의 전개식으로 나타내면,

$$a_m\times n^m+a_{m-1}\times n^{m-1}\cdots+a_1\times n^1+a_0\times n^0$$

로 나타낼 수 있다.

십육진법에서는 16개의 숫자 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F를 사용하고 A, B, C, D, E, F는 각각 10, 11, 12, 13, 14, 15의 값을 나타낸다.

나. 진법의 변환

n진법을 십진법으로 변환하려면 정수부분과 소수부분에 관계없이 n진법의 전개식을 이용하면 된다. 예를 들어,

$$\begin{aligned}765.4_{(8)} &= 7\times8^2+6\times8^1+5\times8^0+4\times8^{-1}\\ &= 7\times64+6\times8+5\times1+\frac{4}{8}=501.5\end{aligned}$$

이고

$$\mathrm{EF}_{(16)}=14\times16^1+15\times16^0=14\times16+15\times1=239$$

이다.

두 번째로, 십진법을 n진법으로 변환하는 경우가 있다. 이 경우에는 정수부분과 소수부분을 분리해서 변환하여 더하는 것이 편리하다. 먼저 십진법의 정수부분을 n진법으로 변환하는 방법을 알아보자. 정수부분은 십진법의 수 N을 n으로 나누어 그 몫을 Q_1이라고 하고 나머지를 R_1이라고 한다. 이때 R_1은 가장 아랫자리 수가 된다. 다시 Q_1을 n으로 나누어 그 몫을 Q_2이라고 하고 나머지를 R_2라고 한다. R_2는 n진법으로 나타낸 수의 다음 자리수가 된다. 계속해서 Q_m이 0일 때까지 나누어서 최종 나머지 R_m이 제일 윗 자리수가 되어 $R_m\cdots R_2R_1$ 순으로 적으면 n진법으로 나타낼 수 있다. 예를 들어 24를 이진법으로 변환해 보자.

$$\begin{array}{r|l l l}
n & \underline{\;N\;} & & \\
n & \underline{\;Q_1\;} & \cdots & R_1 \\
n & \underline{\;Q_2\;} & \cdots & R_2 \\
 & \;\vdots & & \\
n & \underline{\;Q_{m-1}\;} & \cdots & R_{m-1} \\
 & \;0 & \cdots & R_m
\end{array}\;\uparrow
\qquad\qquad
\begin{array}{r|l l l}
2 & \underline{\;24\;} & & \\
2 & \underline{\;12\;} & \cdots & 0 \\
2 & \underline{\;6\;} & \cdots & 0 \\
2 & \underline{\;3\;} & \cdots & 0 \\
2 & \underline{\;1\;} & \cdots & 1 \\
 & \;0 & \cdots & 1
\end{array}\;\uparrow$$

따라서 $24=11000_{(2)}$이다.

예제Ⅲ.1.1

팔진법은 8개의 숫자 0부터 7까지를 사용한다. 그렇다면 $234_{(8)}$ 부터 $456_{(8)}$ 까지의 팔진법의 수는 모두 몇 개인가? 단, $234_{(8)}$와 $456_{(8)}$은 둘 다 포함한다. (2002 고등)

① 146 ② 147 ③ 176 ④ 222 ⑤ 223

<풀이> ②

$234_{(8)}$을 팔진법의 전개식으로 나타내면 $234_{(8)}=156$이 되고 $456_{(8)}$을 팔진법의 전개식으로 나타내면 $456_{(8)}=302$이므로 이 사이의 수는 $302-156+1=147$개 이다.

문제Ⅲ.1.1

다음 이진법 나눗셈의 결과는? (2002 중등)

$$10100001_{(2)} \div 111_{(2)} = ?$$

① $10101_{(2)}$ ② $10111_{(2)}$ ③ $11001_{(2)}$
④ $11010_{(2)}$ ⑤ $11011_{(2)}$

<풀이> ②

이진법의 전개식을 이용하여 십진법의 수로 나타낸다.
$10100001_{(2)} \div 111_{(2)} = 161 \div 7 = 23 = 10111_{(2)}$이 된다.

십진법에서 소수부분을 n진법의 수에 해당되는 소수부분으로 변환하려면 나눗셈 대신에 곱셈을 사용하고 소수점 앞으로 나오는 정수를 가져오면 된다. 즉, 소수부분이 M이라고 하면 M에 n을 곱해서 새로운 정수와 소수를 얻는데, 이때, 나오는 정수를 취하고 남은 소수부분에 n을 곱해서 또 다른 새로운 정수와 소수를 얻는다. 십진법의 수 0.8125를 이진법, 팔진법과 십육진법으로 변환하여 보자.

	계산 결과	새로운 정수	새로운 소수
0.8125 × 2	1.6250	1	0.6250
0.6250 × 2	1.2500	1	0.2500
0.2500 × 2	0.5000	0	0.5000
0.5000 × 2	1.0000	1	0.0000

$0.8125 = 0.1101_{(2)}$ 이고

	계산 결과	새로운 정수	새로운 소수
0.8125 × 8	6.5000	6	0.5000
0.5000 × 8	4.0000	4	0.0000

$0.8125 = 0.64_{(8)}$ 이며,

	계산 결과	새로운 정수	새로운 소수
0.8125 × 16	13.0000	13	0.0000

$0.8125 = 0.\mathrm{D}_{(16)}$ 이다.

정수와 소수로 구성된 십진법의 수를 n진법의 수로 변환할 때에는 정수부분과 소수부분을 따로 계산하여 결합한다. 예를 들어 19.625를 이진법의 수로 변환하면 $19 = 10011_{(2)}$ 이고 $0.625 = 0.101_{(2)}$ 이므로

$$19.625 = 10011.101_{(2)}$$

이 된다.

예제Ⅲ.1.2

이진법의 수 1101.11을 십진법의 수로 표시하면? (2005 교원)

① 11.75　② 13.55　③ 13.75　④ 15.3　⑤ 16.01

<풀이> ③

이진법의 전개식을 사용하면,

$$1101.11_{(2)} = 1\times 2^3 + 1\times 2^2 + 1\times 1 + 1\times \frac{1}{2^1} + 1\times \frac{1}{2^2} = 13.75 \text{ 이다.}$$

문제Ⅲ.1.2

소수 13.875를 이진법의 수로 올바르게 변환한 것은? (2007 교원)

① $1101.111_{(2)}$ ② $1110.111_{(2)}$ ③ $1011.110_{(2)}$
④ $1101.011_{(2)}$ ⑤ $1110.110_{(2)}$

<풀이>

$13 = 1101_{(2)}$ 이고 $0.875 = 0.111(2)$ 이므로 $13.875 = 1101.111_{(2)}$ 이다.

이진법, 팔진법, 십육진법을 상호 변환하여 표시하는 것은 디지털 컴퓨터에서 중요한 일이다. 이진법의 수를 팔진법의 수로 변환할 때, **소수점을 기준으로** 정수부분은 왼쪽으로, 소수부분은 오른쪽으로 각각 이진법의 수를 3개씩 묶어서 계산한다. $110101_{(2)}$ 에서 3개씩 묶으면 $(110)(101)_{(2)}$ 이 되고 각 괄호안의 수를 팔진법의 수로 바꾸면 $65_{(8)}$ 가 된다.

마찬가지로 이진법의 수를 십육진법의 수로 바꾸려면 이진법의 수의 **소수점을 기준으로** 4개씩 묶어서 계산한다. 예를 들어, $110101_{(2)}$ 을 4개씩 묶으면 $(11)(0101)_{(2)} = (0011)(0101)_{(2)}$ 이 되고 각 괄호안의 수를 십육진법으로 변환하면 $35_{(16)}$ 가 된다.

이진법	110	101
팔진법	6	5

이진법	11	0101
십육진법	3	5

팔진법이나 십육진법의 수를 이진법의 수로 변환하려면 팔진법의 수 한 자리가 이진법의 수 3자리로 변환되고 십육진법의 수의 한 자리가 이진법에서 4자리에 해당한다. 팔진법의 수를 십육진법의 수로 변환하려면 이진법을 매개로 하면 편리하다. 즉, 팔진법의 수를 먼저 이진법의 수로 변환한 후에 4자리씩 묶어서 십육진법의 수로 변환하면 된다.

예제Ⅲ.1.3

이진법의 수를 팔진법의 수로 바꿀 때는 $2^3 = 8$ 이므로 이진법의 수를 끝에서부터 세 자리씩 묶어주면 팔진법의 수로 쉽게 바꿀 수 있다. 예를 들어, 이진법의 수 $10101011_{(2)}$ 을 팔진법의 수로 바꾸어 주면 다음과 같이 $253_{(8)}$ 이 된다.

이진법	1 0	1 0 1	0 1 1
팔진법	2	5	3

이와 같은 성질을 이용하여 십육진법의 수 $ABC_{(16)}$을 이진법의 수로 바꾸었을 때, 1의 개수는 몇 개인가? (2004 중등)

① 6개　② 7개　③ 8개　④ 9개　⑤ 10개

<풀이> ②

십육진법의 수	A	B	C
이진법의 수	1010	1011	1100

문제Ⅲ.1.3

팔진법의 수 76을 십육진법의 수로 바꾸면? (2004 교원)

① 3E　② 3D　③ 3C　④ 3B　⑤ 3A

<풀이> ①

팔진법의 수	7	6
이진법의 수(세자리씩)	111	110
이진법의 수(네자리씩)	11	1110
십육진법의 수	3	E

예제Ⅲ.1.4

십육진법의 수 $3FA_{16)}$와 팔진법의 수 $1770_{(8)}$의 관계를 바르게 기술한 것은? (2004 교원)

① $3FA_{16)}$는 $1770_{(8)}$과 같다.　② $3FA_{16)}$는 $1770_{(8)}$보다 2만큼 크다.
③ $3FA_{16)}$는 $1770_{(8)}$보다 2만큼 작다.　④ $3FA_{16)}$는 $1770_{(8)}$보다 1만큼 크다.
⑤ $3FA_{16)}$는 $1770_{(8)}$보다 1만큼 작다.

<풀이> ②

$3FA_{(16)} = 1111111010_{(2)}$이고 $1770_{(8)} = 1111111000_{(2)}$이므로 $3FA_{(16)}$이 $1770_{(8)}$보다 2가 크다.

〈별해〉 $3FA_{(16)} = 3\times16^2 + 15\times16^1 + 10\times16^0 = 768+240+10=1018$ 이고

$1770_{(8)} = 1\times8^3 + 7\times8^2 + 7\times8^1 + 0\times8^0 = 512+448+56=1016$ 이다.

따라서 $3FA_{(16)}$ 가 $1770_{(8)}$ 보다 2만큼 크다.

문제Ⅲ.1.4

이진법의 수 01100011을 팔진법의 수로 바꾸면 얼마인가? (2005 교원)

① 142 ② 143 ③ 144 ④ 145 ⑤ 146

<풀이>

$01100011_{(2)} = 143_{(8)}$ 이 된다.

2진수	01	100	011
8진수	1	4	3

예제Ⅲ.1.5

십육진법의 수 $AD.2F_{(16)}$ 을 팔진법의 수로 변환하시오.

<풀이>

십육진법의 수를 먼저 이진법의 수로 바꾼 다음, 팔진법의 수로 다시 바꾼다. $AD.2F_{(16)} = 10101101.00101111_{(2)}$ 이므로 $255.136_{(8)}$가 된다.

십육진법	A		D	.	2		F
이진법	1010		1101	.	0010		1111
이진법	10	101	101	.	001	011	110
팔진법	2	5	5	.	1	3	6

문제Ⅲ.1.5

팔진법의 수 $756.43_{(8)}$ 을 십육진법의 수로 변환하시오.

<풀이>

팔진법의 수를 먼저 이진법의 수로 바꾼 다음, 십육진법의 수로 다시 바꾼다. $756.43_{(8)} = 111101110.100011_{(2)}$ 이므로 $255.136_{(8)}$가 된다.

다. 보수

과학 계산에 사용되는 수들은 부호, 수의 크기와 소수점으로 표현된다. 여기서 부호는 그 수가 양인지, 음인지를 나타내며 소수점의 위치는 그 수를 표현하는데 필요하다. 수의 부호는 +, −를 가지는데, +는 0으로 −는 1로 표시한다. 보수는 컴퓨터에서 음수 연산을 편리하게 하기 위해 사용하는 것으로 n진법의 수 n에 대한 보수는 n의 보수와 $n-1$의 보수가 있다. 즉, 십진법의 수에는 10의 보수와 9의 보수가 있고 이진법의 수에서는 2의 보수와 1의 보수가 있다.

1) $n-1$의 보수

n진법에서 어떤 수 N의 $n-1$의 보수는 $n-1$에서 N의 각 자리수를 빼면 된다. 예를 들어 십진법의 수인 경우 $10-1=9$가 되고 이진법의 수인 경우는 $2-1=1$이 된다. 즉, 십진법의 수 845의 9의 보수는 $999-845=154$이다. 또, 이진법의 수 $11010_{(2)}$의 1의 보수는 $11111_{(2)}-11010_{(2)}=00101_{(2)}$이다. 이진법의 수에서 1의 보수는 1은 0으로, 0은 1로 바꾸어 놓은 것과 같다. 팔진법의 수에서 7의 보수는 7로부터 각 자리수를 빼고 십육진법의 수에서는 F의 보수는 십진법의 수 15에 해당하는 F에서 각 자리수를 빼면 얻을 수 있다.

예제Ⅲ.1.6 십진법의 수 1234의 9의 보수를 구하시오.

<풀이>

$9999-1234=8765$이다.

문제Ⅲ.1.6 $110101110_{(2)}$에서 1의 보수를 구하시오.

<풀이>

1의 보수는 1은 0으로 0은 1로 바꾸는 것과 같으므로, $001010001_{(2)}$이다.

2) n의 보수

n진법의 수 N의 n의 보수는 $n-1$의 보수를 구한 다음 가장 아래 자리에서 1을 더하면 얻어진다. 예를 들어 십진법의 수 845의 9의 보수가 154이므로 10의 보수는 $154+1=155$가 된다. $11010_{(2)}$의 1의 보수는 $00101_{(2)}$이므로 2

의 보수는 $00101_{(2)}+1_{(2)}=00110_{(2)}$ 이다.

다른 방법으로 n진법의 수 N의 자릿수가 m일 때, n의 보수는 $n^m-\mathrm{N}$이다. 예를 들어, 십진법의 수 5243은 4자리수이므로 m이 4가 되고 10의 보수는 $10^4-5243=4757$이다. 5243.326의 10의 보수는 $10^4-5243.326=4756.674$이다. 마찬가지로, 이진법의 수 $11010_{(2)}$은 5자리수이므로 $100000_{(2)}-11010_{(2)}=00110_{(2)}$이 된다.

예제Ⅲ.1.7

십진법의 수 6738의 10의 보수를 구하시오.

<풀이>

$10^4-6738=3262$이다.

문제Ⅲ.1.7

이진법의 수 $10110_{(2)}$에서 2의 보수를 구하시오.

<풀이>

$2^5-10110_{(2)}=100000_{(2)}-10110_{(2)}=01010_{(2)}$이다.

예제Ⅲ.1.8

십육진법의 수 $\mathrm{ABC}_{(16)}$의 15의 보수와 16의 보수를 구하시오.

<풀이>

15의 보수를 구하면 $\mathrm{FFF}_{(16)}-\mathrm{ABC}_{(16)}=543_{(16)}$이고
16의 보수를 구하면 $16^3-\mathrm{ABC}_{(16)}=1000_{(16)}-\mathrm{ABC}_{(16)}=544_{(16)}$이다.

문제Ⅲ.1.8

팔진법의 수 $567_{(8)}$의 7의 보수와 8의 보수를 구하시오.

<풀이>

7의 보수를 구하면 $777_{(8)}-567_{(8)}=210_{(8)}$이고 8의 보수를 구하면

$8^3-567_{(8)}=1000_{(8)}-567_{(8)}=211_{(8)}$이다.

연 습 문 제

1. 다음 연산의 결과를 십진법의 수로 나타내시오.

$$AB9_{(16)} \div 55_{(8)}$$

① 58 ② 59 ③ 60 ④ 61 ⑤ 62

2. 십육진법의 수 $2BC_{(16)}$을 이진법의 수로 바꾸었을 때 0의 개수는 몇 개인가?

① 2개 ② 3개 ③ 4개 ④ 5개 ⑤ 6개

3. 십진법의 수 172를 십육진법의 수로 바꾸면?

① $AB_{(16)}$ ② $AC_{(16)}$ ③ $AD_{(16)}$ ④ $AE_{(16)}$ ⑤ $AF_{(16)}$

4. 이진법의 수 $10011.011_{(2)}$을 십진법의 수로 표시하면?

① 18.75 ② 19.125 ③ 19.375 ④ 19.875 ⑤ 20.375

5. 십진법의 수 24.625를 이진법의 수로 표시하면?

① $1100.11_{(2)}$ ② $11000.11_{(2)}$ ③ $11001.11_{(2)}$
④ $1100.011_{(2)}$ ⑤ $11000.101_{(2)}$

6. 이진법의 수 $101100011_{(2)}$을 팔진법과 십육진법의 수로 바꾸면?

① $163_{(8)}$, $543_{(16)}$ ② $543_{(8)}$, $163_{(16)}$ ③ $162_{(8)}$, $542_{(16)}$
④ $542_{(8)}$, $162_{(16)}$ ⑤ $544_{(8)}$, $164_{(16)}$

7. 이진법의 수 $10110_{(2)}$에서 2의 보수를 구하면?

① $01001_{(2)}$ ② $10110_{(2)}$ ③ $11111_{(2)}$ ④$00001_{(2)}$ ⑤ $01010_{(2)}$

풀 이

1 ④

$(10\times 16^2+11\times 16^1+9)\times 16^0)\div(5\times 8^1+5\times 8^0)=2745\div 45=61$이다.

2 ③

$2BC_{(16)}=1010111100_{(2)}$이다.

3 ②

$172=AC_{(16)}$이다.

4 ③

$1\times 2^4+1\times 2^1+1+1\times\dfrac{1}{2^2}+1\times\dfrac{1}{2^3}=19+0.25+0.125=19.375$이다.

5 ⑤

$24=11000_{(2)}$이고 $0.625=0.101_{(2)}$이다. 따라서 $24.625=11000.101_{(2)}$

6 ②

$101100011_{(2)}=543_{(8)}=163_{(16)}$이다.

7 ⑤

$2^5-10110_{(2)}=100000_{(2)}-10110_{(2)}=01010_{(2)}$ 이다.

2. 계수

우리가 다루는 대상에 얼마나 많은 경우의 수가 존재하는가하는 물음에 답하기 위해서 실제로 세어보지 않고 순열이나 조합 등의 개념을 이용해서 논리적으로 경우의 수를 구할 수 있다. 순열이나 조합은 확률이나 알고리즘 분석의 기본이 된다. 유한 집합의 원소들 중에서 순서에 따라 나열하는 방법의 수를 **순열**(permutation)이라 하고 후자처럼 순서를 고려하지 않고 선택하는 방법의 수를 **조합**(combination)이라고 한다. 비둘기 집의 원리는 물건을 상자 속에 넣을 때 물건의 수가 상자의 수보다 많으면 반드시 두 개 이상의 물건을 포함하는 상자가 존재한다는 것이다.

가. 순열

서로 다른 원소의 집합에서 순열은 그 원소들의 순서에 따라 나열한 배열이다. 10명의 학생들이 있을 때 4명을 선택하여 줄을 세우는 경우를 생각해 보자. 첫 번째 자리에 줄을 설 수 있는 학생은 10명이고 두 번째 자리에 줄을 설 수 있는 학생은 이미 첫 번째 자리에 줄을 선 한 명의 학생은 제외하므로 9명, 같은 학생이 두 번 설 수 없으므로 다음 자리에 대해서는 8명, 마지막 자리에 대해서는 7명이 올 수 있다. 따라서 순열의 개수는 10・9・8・7이다.

n개의 서로 다른 원소로부터 r의 서로 다른 원소를 뽑는 순열은 $P(n,r)$이나 ${}_nP_r$이라고 나타낸다. $P(n,r)$은

$$P(n,r) = n(n-1)(n-2)\cdots(n-r+1)$$

이다. 10명 중에서 4명만 일렬로 줄을 세우는 순열은 $P(10,4)$이다.

예제Ⅲ.2.1

두 자리 자연수 중에서 각 자리 숫자가 서로 다른 홀수는 몇 개인가?
(2002 초등)

① 30 ② 35 ③ 40 ④ 45 ⑤ 50

<풀이> ③

두 자리의 수의 개수는 10부터 99이므로 90개이다. 이중 홀수의 개수는

$\frac{1}{2}\times 90=45$ 이고 같은 수의 홀수인 11, 33, 55, 77, 99를 빼면 40개이다.

〈곱의 법칙〉 사건 A가 일어날 경우가 m가지, 그 각각에 대하여 사건 B가 일어날 경우가 n가지일 때, 사건 A 또는 사건 B가 동시에 일어날 모든 경우의 수는 $m\times n$(가지)이다.

문제Ⅲ.2.1

우리나라의 일반 전화 국번은 세 자리 또는 네 자리이다. 우리나라에서 개설할 수 있는 전화 국번 중에서 2로 시작하는 국번은 모두 몇 개를 만들 수 있는가?(2005 교원)

① 550 ② 1100 ③ 5500 ④ 11000 ⑤ 15000

<풀이> ②

전화 국번은 세 자리 또는 네 자리이므로 세 자리 국번의 수와 네 자리 국번의 수를 각각 구한 후 합의 법칙으로 더하면 된다. 주어진 문제에서 전화 국번은 항상 2로 시작하므로 세 자리 국번은 $\boxed{2}\boxed{}\boxed{}$, 네자리 국번 $\boxed{2}\boxed{}\boxed{}\boxed{}$의 개수를 구하면 된다. 개설할 수 있는 세 자리 국번의 수는 두 번째와 세 번째 자리에 올 수 있는 숫자가 0부터 9까지 총 10개이므로 $10\times 10=100$가지이다. 마찬가지로 네 자리 전화국번의 경우 첫 번째 자리를 제외한 나머지 자리에 올 수 있는 숫자의 경우의 수는 $10\times 10\times 10=1000$가지이다.

따라서 전체 세 자리와 네 자리로 된 전화 국번 중에서 2번으로 시작하는 국번은 $100+1000=1100$이다.

〈합의 법칙〉 사건 A가 일어날 경우가 m가지, 사건 B가 일어날 경우가 n가지가 있고, 두 사건 A, B가 동시에 일어나지 않을 때, 사건 A 또는 사건 B가 일어날 경우의 수는 $m+n$(가지)이다.

n개 중에 같은 것이 $p, q, r, \cdots, s$개가 있을 때, 이 n개를 일렬로 배열하는 순열의 수는

$$\frac{n!}{p!q!,\, r!,\, \cdots,\, s!} \quad (\text{단, } p+q+r+\cdots+s \leqq n)$$

이다.

$P(n,r)$은 팩토리얼(계승, factorial) 함수를 사용하여 계산할 수 있다. n팩토리얼은 $n \cdot (n-1) \cdot (n-2) \cdots 1$로 계산하고 $n!$이라고 나타낸다. $0! = 1$로 정의한다. $n! = n \cdot (n-1)!$이다. $P(10,4)$을 팩토리얼 함수를 사용하여 다시 나타내면,

$$\begin{aligned} P(10,4) &= 10 \cdot 9 \cdot 8 \cdot 7 \\ &= \frac{10 \cdot 9 \cdot 8 \cdot 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1}{6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1} \\ &= \frac{10!}{6!} = \frac{10!}{(10-4)!} \end{aligned}$$

이다.

일반적으로

$$P(n,r) = \frac{n!}{(n-r)!}, \quad (0 \leqq r \leqq n)$$

가 된다.

예제Ⅲ.2.2

$N! = N \times (N-1) \times (N-2) \times \cdots \times 2 \times 1$이다. 예를 들어,

$4! = 4 \times 3 \times 2 \times 1 = 24$가 된다. 그렇다면 $20! \div 18!$의 값은? (2005 초등)

① 320 ② 340 ③ 360 ④ 380 ⑤ 400

<풀이> ④

$$20! \div 18! = \frac{20 \times 19 \times 18 \times 17 \times 16 \times \cdots \times 1}{18 \times 17 \times 16 \times \cdots \times 1} = 20 \times 19 = 380$$

문제Ⅲ.2.2

$(20!)^2$을 계산한 결과에는 끝에 0이 몇 개 붙어 있을까?

(단, $n! = n \times (n-1) \times (n-2) \times \cdots \times 1$ 이다.)(2002 고등)

① 8개 ② 9개 ③ 10개 ④ 11개 ⑤ 12개

<풀이> ①

$(20!)^2$에는 2가 18개, 5가 8개 있고 $2\times5=10$이므로 0이 8개가 생긴다.

예제Ⅲ.2.3

문자 a, b, c에서 중복을 허용하여 세 개를 택하여 만든 단어를 전송하려고 한다. 단, 전송되는 단어에 a가 연속이 되면 수신이 불가능하다고 한다. 예를 들어, aab와 aaa 등은 수신이 불가능하고 bba와 aba 등은 수신이 가능하게 된다. 수신 가능한 단어는 몇 개인가?

① 21개 ② 22개 ③ 23개 ④ 24개 ⑤ 25개

<풀이> ②

a, b, c에서 중복을 허락하여 3개를 뽑아 나열하는 방법의 수는 $3^3=27$(가지)이다. 여기서 a가 연속하여 있는 경우를 제외한다.

(1) a가 2개 연속할 때, aab, aac, baa, caa의 4가지이고,

(2) a가 3개 연속할 때, aaa의 1가지이다.

따라서 수신 가능한 단어는 27-4-1=22가지이다.

문제Ⅲ.2.3

영문자 P, A, S, S를 일렬로 배열하는 방법은 몇 가지인가?

① 6개 ② 8개 ③ 12개 ④ 18개 ⑤ 24개

<풀이> ③

S를 2개 포함한 4개의 문자를 일렬로 배열하는 순열의 수이므로 $\frac{4!}{2!}=12$가지이다.

나. 조합

n개의 원소로 이루어진 하나의 집합으로부터 r개의 원소를 순서 없이 선택하는 것을 n개의 서로 다른 원소로부터 선택된 r개의 서로 다른 원소의 조합

이라고 하고 $C(n,r)$이나 ${}_nC_r$ 혹은 $\binom{n}{r}$로 나타낸다. n개의 대상에서 n개를 선택하는 방법의 수는 1이므로 $C(n,n)=1$이고, n개의 대상에서 1개를 선택하는 방법의 수는 n이므로 $C(n,1)=n$이다.

n개의 서로 다른 원소로 이루어진 집합에서 r개의 서로 다른 원소를 선택하는 순열은 먼저 $C(n,r)$을 구한 후 각 $C(n,r)$에서 $r!$가지 방법으로 원소를 나열하는 것과 같다. 그래서 $P(n,r)=C(n,r)\cdot r!$이다. $0 \leq r \leq n$일 때, n개의 서로 다른 원소로 이루어진 집합에서 r개의 서로 다른 원소를 선택하는 조합의 개수는

$$C(n,r) = \frac{P(n,r)}{r!} = \frac{n!}{r!\,(n-r)!}$$

이다.

예제Ⅲ.2.4

1, 2, 3, 4, 5, 6, 7 이렇게 7개의 숫자에서 세 개의 숫자를 차례대로 선택한다. 이 때 첫 번째로 선택된 수는 두 번째로 선택된 수보다 작아야 하고, 세 번째로 선택된 수는 첫 번째로 선택된 수와 두 번째로 선택된 수의 합보다 커야 한다. 이와 같이 숫자를 선택하는 방법은 모두 몇 가지인가? (2002 중등)

① 11개 ② 12개 ③ 13개 ④ 14개 ⑤ 15개

<풀이> ③

수형도를 이용하여 개수를 구해보면 다음과 같다.

(1 2 4) (1 2 5) (1 3 5) (1 2 6) (1 3 6) (1 4 6) (2 3 6)
(1 2 7) (1 3 7) (1 4 7) (1 5 7) (2 3 7) (2 4 7)

따라서 13개이다.

문제Ⅲ.2.4

여덟 개의 숫자 1, 2, 3, 4, 5, 6, 7, 8에서 세 개의 숫자 또는 네 개의 숫자를 차례대로 선택한다. 이 때 첫 번째로 선택된 수는 두 번째로 선택된 수보다 작아야 하고, 세 번째로 선택된 수와 네 번째로 선택된 수는 바로 앞서 선택된 두 숫자의 합보다 커야 한다. 이와 같이 숫자를 선택하는 방법은 모두 몇 가지인가? (2002 고등)

① 21개 ② 22개 ③ 23개 ④ 24개 ⑤ 25개

<풀이> ⑤

수형도를 이용하여 개수를 구해보면,

(1 2 4) (1 2 5) (1 2 6) (1 2 7) (1 3 5) (1 3 6) (1 3 7) (1 4 6)
(1 4 7) (1 5 7) (2 3 6) (2 3 7) (2 4 7) (1 2 8) (1 3 8) (1 4 8)
(1 5 8) (1 6 8) (2 3 8) (2 4 8) (2 5 8) (3 4 8) ·································· 22개
(1 2 4 7) (1 2 4 8) (1 2 5 8) ·· 3개

따라서 22+3=25개이다.

예제Ⅲ.2.5

$n>2$일 때, $C(n)=2C(n/2)+2$이고 $C(2)=1$, $C(1)=1$이다. $n=2k, k\geq 0$일 때 $C(n)$의 값은 얼마인가?(2004 교원)

① $1.5n-2$ ② $n+2$ ③ $2n-1$ ④ $2n-2$ ⑤ 답 없음

<풀이> ①

$C(4)=2C(2)+2=4$

$C(8)=2C(4)+2=8+2=10$

$C(16)=2C(8)+2=20+2=22$

따라서 $C(n)=1.5n-2$ 이다.

문제Ⅲ.2.5

n개의 서로 다른 원소를 가진 집합 $X=\{x_1, x_2, \cdots, x_n\}$에서 X의 r조합은 X에서 순서에 상관없이 r개의 원소를 선택하는 것이다. 또한, $X=\{1, 2, 3, \cdots, n\}$의 r조합은 $X=\{s_1, s_2, \cdots, s_r\}$을 만족하는 숫자열 $s_1<s_2<\cdots<s_r$을 만족하는 숫자열 $s_1 s_2 \cdots s_r$로 표현하면 r조합을 순서대로 나열할 수 있다. 그러면, $\{1, 2, 3, 4, 5, 6, 7\}$의 5조합의 첫 번째 숫자열은 12345이고 마지막 숫자열은 34567이 된다. $\{1, 2, 3, 4, 5, 6, 7\}$의 5조합을 나열할 때 12367바로 다음에 올 숫자열은 무엇인가?(2004 교원)

① 12376 ② 12435 ③ 12456 ④ 12467 ⑤ 13267

<풀이> ③
12367에서 마지막 숫자열이 67이므로 그 앞의 숫자가 1증가하게 된다. 따라서 12456이 된다.

서로 다른 n개의 대상으로부터 중복을 허용하여 r개를 선택하는 것을 중복조합이라고 하고 ${}_nH_r$로 나타낸다. 예를 들어, 1, 2, 3의 세 수에서 중복을 허용하여 두 수를 뽑는 조합의 수를 구한다고 하면, 11, 12, 13, 22, 23, 33의 여섯 가지이다.

일반적으로 서로 다른 n개에서 중복을 허락하여 r개를 뽑는 조합의 수 ${}_nH_r$은 서로 다른 $(n+r-1)$개에서 중복을 허락하지 않고 r개를 택하는 조합의 수 $C(n+r-1,\ r)$ 와 같다. 즉

$$ {}_nH_r \ = \ C(n+r-1,\ r) $$

위의 예를 다시 풀어보면, 서로 다른 3개에서 중복을 허락하여 2개를 뽑는 조합의 수이므로 ${}_3H_2 = \ C(3+2-1,\ 2) = \ C(4,\ 2) = 6$이다.

예제Ⅲ.2.6

서로 다른 두 개의 숫자 카드에서 중복을 허락하여 4개를 뽑는 조합의 수를 구하여라.

<풀이>

$$ {}_2H_4 = C(2+4-1,4) = C(5,4) = C(5,1) = 5 $$

문제Ⅲ.2.6

과자 가게에서 네 가지 종류의 과자를 팔고 있다. 여섯 개의 과자를 뽑는 조합의 수를 구하여라.

<풀이>

$$ {}_4H_6 = C(4+6-1,6) = C(9,6) = C(9,3) = 84 $$

다. 비둘기 집의 원리

비둘기들이 날아서 비둘기 집으로 들어간다고 하자. 비둘기 집의 원리(pigeonhole principle)는 비둘기 집보다 비둘기가 더 많을 경우 적어도 한 개의 비둘기 집에는 두 마리 이상의 비둘기가 들어가게 된다는 것을 말한다.

즉, $(n+1)$마리의 비둘기와 n개의 비둘기 집이 있을 때 적어도 한 집은 두 마리의 비둘기를 갖는다. 만약 n개의 각 집이 기껏해야 한 마리의 비둘기를 가진다면, 비둘기의 전체 수는 기껏해야 n마리가 된다. 예를 들어, 13명의 학생이 있으면 적어도 두 명은 태어난 달이 같아야 한다.

비둘기 집의 원리를 확장한다면 $(2n+1)$마리의 비둘기가 n개의 집에 들어갈 때 적어도 한 개의 집은 세 마리의 비둘기를 갖게 된다. $(3n+1)$마리의 비둘기가 n개의 집에 들어갈 때 적어도 한 개의 집은 네 마리의 비둘기를 갖게 된다. 일반화된 비둘기 집의 원리는 다음과 같다.

m마리의 비둘기가 n개의 비둘기 집에 들어간다면 $\left\lfloor \frac{m-1}{n} \right\rfloor +1$마리 이상 들어간 비둘기집이 적어도 하나 있다.

$\left\lfloor \frac{m}{n} \right\rfloor$은 $\frac{m}{n}$보다 작거나 같은 정수 중에서 가장 큰 정수이다. 예를 들면, $\left\lfloor \frac{3}{2} \right\rfloor = 1$, $\left\lfloor \frac{19}{4} \right\rfloor = 4$이고 $\left\lfloor \frac{15}{5} \right\rfloor = 3$이다.

100명의 사람이 있으면 그 중에서 적어도 9명은 같은 달에 태어났다. 비둘기 집의 원리를 사용하려면 주어진 문제에서 비둘기가 무엇인지와 비둘기 집이 무엇인지를 확인하고 비둘기 수와 비둘기 집의 수를 셀 수 있어야 한다. 비둘기 집의 원리는 두 마리 이상 들어간 비둘기 집이 있는지 없는지에 대한 판단을 내릴 수 있게 도와주지만, 어떤 비둘기 집에 비둘기가 더 들어갔는지에 대한 판단을 내리는데 도움을 주지는 않는다.

예제Ⅲ.2.7

26개의 빨간색 구슬과 26개의 파란색 구슬이 있다. 보지 않고 구슬을 하나씩 고른다고 할 때, 최소한 몇 번을 골라야 빨간색 구슬을 찾아낼 수 있다고 보장할 수 있을까?(2002 중등)

① 21 ② 25 ③ 27 ④ 30 ⑤ 33

<풀이> ③

26개의 빨간색 구슬과 26개의 파란색 구슬이 있다면 26번까지 모두 파란색 구슬만 나올 수 있다. 그러므로 최소한 27번이 되어야 빨간색 구슬을 찾아낼 수 있다.

문제Ⅲ.2.7

0에서 100까지의 숫자가 적힌 카드가 있다. 한 장의 카드에는 한 개의 수가 적혀 있고, 어떤 두 카드에도 같은 숫자가 적혀 있지 않다. 카드를 한 장씩 뽑는다고 할 때, 최소 몇 장의 카드를 뽑아야 홀수가 적힌 카드를 적어도 한 장 뽑을 수 있다고 보장할 수 있는가? (2003 중등)

① 50번　② 51번　③ 52번　④ 53번　⑤ 100번

<풀이> ③

0에서 100까지 짝수가 모두 51개가 있으므로 처음부터 51번까지 뽑은 카드가 모두 짝수였다고 가정하더라도 최소 52번을 뽑으면 홀수가 적힌 카드를 뽑을 수 있다.

예제Ⅲ.2.8

옷장 서랍에 10개의 검은색 양말과 10개의 흰색 양말이 있다. 서랍 안을 보지 않고 손을 넣어 임의로 양말을 꺼낼 때 최소한 몇 개의 양말을 꺼내야 색깔이 같은 한 쌍의 양말을 꺼낼 수 있다고 보장할 수 있는가?(2003 중등)

① 3개　② 5개　③ 7개　④ 9개　⑤ 11개

<풀이> ①

양말을 두 개 꺼낼 경우 흰색과 흰색이나 검정색과 검정색은 이미 색깔이 같은 한 쌍의 양말을 꺼낸 경우이다. 흰색과 검정을 꺼낼 때, 세 번째 꺼낸 양말은 흰색이 아니면 검정색이므로 최소한 3개의 양말을 꺼내면 색깔이 같은 한 쌍의 양말이 존재하게 된다.

문제Ⅲ.2.8

축구 교실에 참가한 학생들에게 30이하의 짝수로 등번호를 주려고 한다. 서로 같은 등번호인 학생들이 반드시 존재하도록 하려면 적어도 몇 명의 학생이 참가해야 하는가?

① 14명　② 15명　③ 16명　④ 29명　⑤ 31명

<풀이> ③

번호를 최대한 다르게 준다고 할 경우 15명의 학생에게 $2, 4, 6, \cdots, 30$의 번호를 줄 수 있다. 따라서 한 명만 더 들어오면 기존에 주었던 짝수 중에 하나를 등번호로 주어야 하므로 서로 같은 등번호인 학생들이 반드시 존재하게 된다. 그러므로 서로 같은 등번호인 학생들이 반드시 존재하려면 적어도 16명이 있어야 한다.

예제Ⅲ.2.9

26명의 학생들을 대상으로 생일을 조사하였다. 이들 중에는 생일이 같은 달에 있는 사람이 적어도 몇 명인가?

① 3명 ② 4명 ③ 5명 ④ 6명 ⑤ 7명

<풀이> ①

$\left\lfloor \frac{25}{12} \right\rfloor + 1 = 3$이므로 생일이 같은 달에 있는 사람은 적어도 3명이다.

문제Ⅲ.2.9

한 변의 길이가 3인 정사각형의 내부에 10개의 점을 임의로 찍을 때, 이들 점 사이의 거리에 대한 설명으로 항상 옳은 것은?

① 임의의 두 점 사이의 거리는 $\sqrt{2}$이다.
② 임의의 두 점 사이의 거리는 $\sqrt{3}$이다.
③ 두 점 사이의 거리 중에서 $\sqrt{2}$이하인 것이 반드시 있다.
④ 두 점 사이의 거리 중에서 $\sqrt{3}$이하인 것이 반드시 있다.
⑤ 위에는 정답이 없다.

<풀이> ③ 한 변의 길이가 1인 9개의 정사각형으로 나누면 대각선의 길이는 $\sqrt{2}$가 되고 10개의 점을 찍으면 어느 한 정사각형에는 두 개의 점이 존재하게 된다. 그러므로 두 점 사이의 거리가 $\sqrt{2}$이하인 것이 반드시 존재한다.

연 습 문 제

1. 5명의 사람들 중에서 회장과 부회장을 뽑는 방법은 몇 가지인가?

① 16가지 ② 18가지 ③ 20가지 ④ 24가지 ⑤ 25가지

2. 7비트의 이진법의 수 가운데 1111로 시작하는 이진법의 수는 몇 개인가?

① 2개 ② 4개 ③ 8개 ④ 16개 ⑤ 32개

3. 6개의 0과 8개의 1로 구성된 비트 문자열 중에서 모든 0은 1 바로 다음에 위치하는 것은 모두 몇 개인가?

① 28개 ② 30개 ③ 32개 ④ 34개 ⑤ 36개

4. 숫자 1, 2, 3, 4, 5를 이용하여 중복을 허용하지 않고 두 자리 홀수를 만드는 방법은 몇 가지가 있는가?

① 9개 ② 10개 ③ 12개 ④ 15개 ⑤ 20개

5. 비둘기 집의 원리를 일반화하면 N개의 물건을 M개의 상자에 넣는다면 적어도 하나의 상자에는 $\left\lfloor \frac{N-1}{M} \right\rfloor$+1개의 물건이 들어가게 된다. 이 내용을 바탕으로 200명의 사람이 있을 때 같은 달에 태어난 사람은 최소 몇 명이나 되는가?

① 16명 ② 17명 ③ 18명 ④ 19명 ⑤ 20명

6. 0과 1로만 이루어진 다섯 자리 문자열 중에서 문자열 '111'을 포함하지 않는 문자열의 개수는?

① 23개 ② 24개 ③ 25개 ④ 26개 ⑤ 27개

풀 이

1

③

5명의 사람들 중에서 2명을 순서대로 선택하는 방법의 수는 $P(5,2)=5\times4=20$가지이다.

2

③

7비트 중에서 처음의 4비트가 1111로 고정되어 있으므로 뒤의 3비트만 선택할 수 있다. 5번째 비트에 0또는 1을 선택하고 그 각각에 대하여 6번째 비트에 0또는 1을 선택할 수 있고 다시 각각에 대하여 7번째 비트에도 0 또는 1을 선택할 수 있다. 따라서 곱의 법칙에 따라 $2\times2\times2=8$개다.

3

①

6개의 0과 8개의 1로 구성된 비트 문자열 중에서 모든 0은 1의 바로 다음에 위치하므로 01을 X로 둔다면 8개 중에서 X가 6개, 1이 2개 있고, 이 8개를 일렬로 배열하는 순열의 수는 $\frac{8!}{6!\cdot2!}=28$이다.

4

③

홀수가 되어야 하므로 일의 자리에 올 수 있는 숫자는 1, 3, 5이고 두 자리 수이므로 십의 자리에 올 수 있는 각각 4가지이므로 $4\times3=12$가지이다.

5

②

200명의 사람이 있으면 그 중에서 적어도 $\left\lfloor\frac{199}{12}\right\rfloor+1=\lfloor16.67\rfloor+1=17$ 명은 같은 달에 태어났다.

6

②

0과 1로만 만들 수 있는 다섯자리 문자열의 경우의 수는 $2^5=32$이다. 이 중에서 연속하는 111을 포함하는 문자열은 8개로 00111, 01110, 01111, 10111, 11110, 11100, 11101, 11111이다. 따라서 $32-8=24$이다.

3. 관계

곱집합의 부분집합인 관계는 집합의 원소들 간의 순서를 고려할 경우에 사용된다. 두 집합의 원소들 간의 관계를 표현하는 가장 직접적인 방법은 순서쌍을 이용하는 것이다. 실생활에서의 사람과 사람의 관계에는 부모와 자식간의 관계, 교사와 학생사이의 관계, 생일이 같은 관계 등 무수히 많다. 일반적으로 n개의 객체간의 관계를 n항관계라고 하며, 보통 관계라고 하면 이항관계(binary relation)를 의미한다.

가. 관계

두 집합 A, B가 있을 때, $a \in A$, $b \in B$에 대한 모든 순서쌍 (a,b)의 집합을 A와 B의 곱집합이라 하고 $A \times B$라고 나타낸다.

$$A \times B = \{(a,b) \mid a \in A,\ b \in B\}$$

A에서 B로의 관계 R은 $A \times B$의 부분집합이다.

A와 B가 공집합이 아닌 집합이고 $a \in A$, $b \in B$라고 하자. 집합 A에서 집합 B로의 관계 R은 $A \times B$의 부분집합이고, $(a,b) \in R$이면 a는 b에 대해 R의 관계가 있다고 하며 aRb로 나타낸다. $(a,b) \not\in R$이면, a는 b에 대해 R의 관계가 없다고 하고 $a \not R b$로 나타낸다. 이때, 관계 R의 모든 순서쌍에서 첫 번째 원소들의 집합을 정의역(domain)이라고 하고 Dom(R)로, 두 번째 원소들의 집합을 치역(range)이라고 하며 Ran(R)로 표현한다.

집합 $A = \{1,2,3\}$, $B = \{1,2\}$라 하면 관계 $R = \{(1,1),(2,2),(3,1)\}$은 $A \times B$의 부분집합이므로 집합 A에서 집합 B로의 관계가 된다. 관계 R이 $A \times B$의 모든 원소로 이루어져 있으면, R은 전체관계(universal relation)라고 하고 $R = \varnothing$이면 공관계(영관계, empty relation)라고 한다. 집합 A에서 집합 A로의 관계는 보통 집합 A에서의 관계라고 말한다.

집합 A에서 집합 B로의 관계 R에 대하여 집합 B에서 집합 A로의 관계를 R의 역관계(inverse relation)라고 하고 R^{-1}로 나타내고

$$R^{-1} = \{(b,\ a) \mid (a,\ b) \in R\}$$

이다.

예를 들어, $A = \{1,2,3\}$에서의 관계 $R = \{(x,y) \mid x$는 y보다 작다$\}$라고 할

때, $R=\{(1,2),(1,3),(2,3)\}$ 이므로 $R^{-1}=\{(2,1),(3,1),(3,2)\}$ 이다.
즉, $R^{-1}=\{(x,y) \mid x$는 y보다 크다$\}$ 이다.

집합 A에서 집합 B로의 관계 R은 $A\times B$의 부분집합일 때, R의 여집합 $\overline{R}$를 여관계(complementary relation)이라 하고

$$\overline{R}=\{(a,b) \mid (a,b)\notin R\}$$

이다.

예제Ⅲ.3.1

$A=\{1,2,3\}$일 때, $a, b\in A$에 대하여 $a\leq b$를 만족하는 관계 R, R^{-1}와 $\overline{R}$을 구하여라.

<풀이>

$$R=\{(1,1),(1,2),(1,3),(2,2),(2,3),(3,3)\}$$
$$R^{-1}=\{(1,1),(2,1),(3,1),(2,2),(3,2),(3,3)\}$$
$$\overline{R}=\{(2,1),(3,1),(3,2)\}$$

문제Ⅲ.3.1

$A=\{1,2,3\}$일 때, $a,b\in A$에 대하여 $R=\{(1,1),(2,2),(3,3)\}$일 경우 R^{-1}와 $\overline{R}$를 구하여라.

<풀이>

$$R^{-1}=\{(1,1),(2,2),(3,3)\}$$
$$\overline{R}=\{(1,2),(1,3),(2,1),(2,3),(3,1),(3,2)\}$$

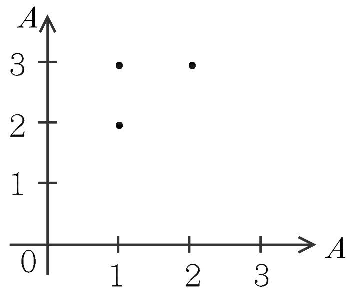
나. 관계의 표현 방법

지금까지는 관계를 원소나열법과 조건제시법으로 표현하였다. 또 다른 관계의 표현 방법에는 좌표도표(coordinate diagram), 화살표도표(arrow diagram), 방향그래프(directed graph, digraph), 관계행렬(relation matrix) 등을 이용하는 방법이 있다.

1) 좌표도표

집합 A의 원소를 x축 위의 점으로, 집합 B의 원소를 y축 위의 점으로 생각하여 aRb이면 좌표평면상에 점 (a,b)로 표시하는 방법을 좌표도표라고 한다.

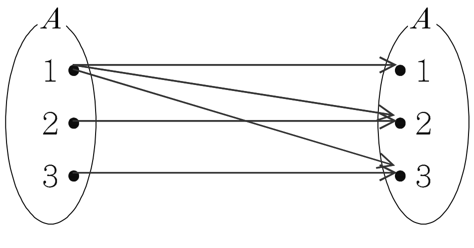
예를 들어, $A=\{1,2,3\}$에서의 R이 a가 b보다 작은 관계라고 할 때, R을 좌표도표로 나타내면, 〔그림Ⅲ.3.1〕과 같다.



[그림Ⅲ.3.1] 좌표도표

2) 화살표도표

A의 원소를 한 구역에, B의 원소를 다른 한 구역에 표시하여 aRb이면 집합 A의 원소 a로부터 집합 B의 원소 b로 화살표를 그려 관계를 나타내는 방법을 화살표도표라고 한다. 예를 들어 $A=\{1,2,3\}$에서의 R이 b는 a로 나누어 떨어지는 관계인 경우, R을 화살표 도표로 표현하면 〔그림Ⅲ.3.2〕와 같다.

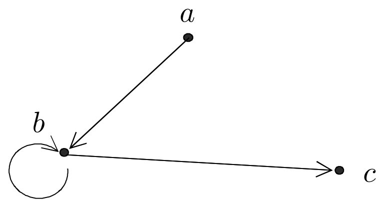


[그림Ⅲ.3.2] 화살표도표

3) 방향그래프

집합 A는 유한 집합이고 관계 R은 집합 A에 대한 관계라면 먼저 집합 A의 각 원소를 그래프의 각 정점(vertex)으로 표시한다. 그 다음에 aRb면 a에서 b까지 간선(edge, 연결선)으로 연결한다. 그러므로 정점은 집합 A의 원소에 대응하며 간선의 개수는 순서쌍의 개수와 같다. 이와 같이 관계 R을 표현하는 방법을 R의 방향그래프 혹은 유향그래프라고 한다.

예를 들어 집합 $A=\{a,b,c\}$에서 관계 $R=\{(a,b),(b,b),(b,c)\}$라면 R의 방향그래프는 〔그림Ⅲ.3.3〕과 같다.



[그림Ⅲ.3.3] 방향그래프

4) 관계행렬

A의 원소를 행으로, B의 원소를 열로 하는 행렬에서 aRb이면 1로, $a\not R b$이면 0으로 나타낸다. 즉, $A=\{a_1,a_2,\cdots,a_m\}$, $B=\{b_1,b_2,\cdots,b_n\}$이며, 관계 R은 집합 A에서 B로의 관계라면 $m\times n$행렬 $M_R=(m_{ij})$에 의해 R을 표현할 수 있다. m_{ij}는 다음과 같이 정의하고 M_R을 R의 관계행렬이라고 한다.

$$m_{ij}=\begin{cases}1 & \text{if } (a_i,b_j)\in R \\ 0 & \text{if } (a_i,b_j)\not\in R\end{cases}$$

예를 들어 $A=\{1,2,3,4\}$에서의 관계 $R=\{(1,2),(1,3),(1,4),(2,3),(2,4),(3,4)\}$의 관계행렬은

$$M_R=\begin{bmatrix}0&1&1&1\\0&0&1&1\\0&0&0&1\\0&0&0&0\end{bmatrix}$$

이다.

다. 관계의 성질

집합 A에서의 관계 R의 성질은 반사관계(reflexive relation), 비반사관계(irreflexive relation), 대칭관계(symmetric relation), 반대칭관계(antisymmetric relation), 추이관계(transitive relation) 등이 있다.

1) 반사관계

A의 모든 원소 a에 대하여 aRa이면, R은 반사관계라고 한다. 즉, $(a,\ a)\in R$이어야 하므로 반사관계 R을 방향그래프로 표현하면 그래프의 모든

정점에서 자기 자신을 가리키는 루프를 가진다. 관계행렬 M_{ij}로 표현하면 주대각행렬 M_{ii}의 원소가 모두 1이 되어야 하고 어느 한 원소라도 0이 되면 반사관계가 성립하지 않는다.

예제Ⅲ.3.2

다음은 집합 $A=\{1,2,3,4\}$에 대한 관계이다. 이들 중에서 반사관계인 것을 찾으시오.

① $R_1=\{(1,1),(1,2),(2,1),(2,2),(3,3),(3,4),(4,3),(4,4)\}$

② $R_2=\{(1,1),(1,2),(1,3),(1,4),(2,2),(2,3),(2,4),(3,3),(3,4),(4,4)\}$

③ $R_3=\{(1,2),(1,3),(1,4),(2,3),(2,4),(3,4)\}$

④ $R_4=\{(1,1),(2,2),(3,3),(4,4)\}$

⑤ $R_5=\varnothing$

⑥ $R_6=\{(1,1),(1,2)\}$

<풀이> ①, ②, ④

반사관계가 되기 위해서는 $(1,1)$, $(2,2)$, $(3,3)$, $(4,4)$의 모든 순서쌍을 포함하고 있어야 한다. 이 중 하나의 순서쌍이라도 없을 경우 반사관계가 성립하지 않는다. 관계 R_3, R_5와 R_6은 반사관계가 성립하지 않는다.

문제Ⅲ.3.2

다음은 집합 $A=\{1,2,3,4\}$에 대한 관계들이다. 이들 중에서 반사관계인 것을 찾으시오.

① $R_7=\{(1,1),(1,2),(2,1),(2,2)\}$

② $R_8=\{(1,1),(1,2),(1,4),(2,2),(2,4),(3,3),(4,4)\}$

③ $R_9=\{(1,1)\}$

④ $R_{10}=\{(x,y)\mid y=x\}$

⑤ $R_{11}=\{(x,y)\mid x+y\leq 4\}$

⑥ $R_{12}=\{(1,3),(3,1)\}$

<풀이> ②, ④

R_7, R_9, R_{11}과 R_{12}는 반사관계가 아니다.

2) 비반사관계

A의 모든 원소 a에 대하여 $a \not R a$이면, R은 비반사관계라고 한다. 즉, $(a, a) \notin R$이어야 되므로 반사관계 R을 방향그래프로 표현하면 그래프의 어떠한 정점에서도 자기 자신을 가리키는 루프를 가지면 안 된다. 관계행렬로 표현하면 주대각행렬의 원소가 모두 0이 되어야 하며 어느 한 원소도 1이 되면 비반사관계가 성립하지 않는다.

예제Ⅲ.3.3

예제Ⅲ.3.2의 어떤 관계가 비반사관계인지를 찾아보시오.

<풀이> ③, ⑤

문제Ⅲ.3.3

문제Ⅲ.3.2의 어떤 관계가 비반사관계인지를 찾아보시오.

<풀이> ⑥

3) 대칭관계

A의 모든 원소에 대하여 aRb일 때 bRa이면 R은 대칭관계라고 한다. 방향그래프에서 정점 a에서 b로의 간선이 존재할 경우 b에서 a로의 되돌아오는 간선이 존재해야만 대칭관계가 된다. 어느 한 간선이라도 되돌아오는 간선이 없다면 대칭관계가 되지 않는다. 관계행렬에서 1이 되는 a_{ij}의 값과 대칭이 되는 a_{ji}의 값에 1이 있어야 대칭관계가 된다.

예제Ⅲ.3.4

예제Ⅲ.3.2의 어떤 관계가 대칭관계인가?

<풀이> ①, ④, ⑤

R_2, R_3과 R_6에서 모두 (1,2)는 있는데 (2,1)이 없으므로 대칭관계가 아니다.

문제Ⅲ.3.4

문제Ⅲ.3.2의 어떤 관계가 대칭관계인가?

<풀이> ①, ③, ④, ⑤, ⑥

R_8에서 (1,2)는 있는데 (2,1)이 없으므로 대칭관계가 아니다.

4) 반대칭관계

A의 모든 원소에 대하여 aRb이고 bRa일 때 $a=b$이면 R은 반대칭관계라고 한다. 즉, $a \neq b$이면 $a\not{R}b$이거나 $b\not{R}a$일 때 반대칭관계가 성립한다. 방향그래프에서 정점 a에서 b로의 간선이 존재할 때 b에서 a로 되돌아오는 간선이 하나도 없다면 반대칭관계가 된다.

관계행렬에서 1이 되는 a_{ij}의 값과 대칭이 되는 a_{ji}값이 어느 하나라도 1이 되지 않아야 반대칭관계가 된다.

예제Ⅲ.3.5

집합 A={x, y, z}에 대한 다음 관계가 대칭관계 및 반대칭관계가 모두 성립하는 것은? (2007 교원)

① R1 = {(x, z)}

② R2 = {(y, z), (z, y)}

③ R3 = {(x, x), (y, y), (z, z)}

④ R4 = {(x, y), (x, z), (y, x), (z, z)}

⑤ R5 = {(x, y), (y, z), (z, x)}

<풀이> ③

① (x, z)만 있으므로 반대칭관계만 성립한다.

② (y, z), (z, y)가 있으므로 대칭관계는 성립하지만 반대칭관계는 성립하지 않는다.

④ (x, y)가 있을 때, (y, x)가 있으므로 대칭관계는 성립하지만 반대칭관계는 성립하지 않는다.

⑤ (x, y)가 있을 때, (y, x)가 없으므로 대칭관계가 성립하지 않는다.

문제Ⅲ.3.5

문제Ⅲ.3.2의 어떤 관계가 반대칭관계인가?

<풀이> R_8, R_9와 R_{10}

R_7는 (1,2)와 (2,1)이 동시에 있으므로 반대칭관계가 아니다.

5) 추이관계

A의 모든 원소에 대하여 aRb이고 bRc일 때 aRc이면, R은 추이관계(전이관계 혹은 이행 관계)라고 한다. aRb이고 bRc이지만 $a\not{R}c$일 때, R은 추이관계가 아니다. 즉, 방향그래프에서 정점 a에서 b로의 간선과 b에서 c로의 간선이 존재할 때 a에서 c로의 간선이 반드시 존재해야만 추이관계가 된다. 만약 정점 a에서 b로의 간선과 b에서 c로의 간선은 존재하지만 a에서 c로의 간선이 존재하지 않는다면 추이관계가 성립하지 않는다.

간선이 하나도 없거나 정점 a에서 b로의 간선만 존재한다면 더 이상 고려할 간선이 없으므로 추이관계가 성립하는 것으로 본다. 대칭, 반대칭, 추이관계는 가정이 불충분한 경우 그 관계가 성립하는 것으로 간주한다.

☆ 주어진 모든 원소 a에 대해 $(a,a)\in A$이면 반사관계
☆ 주어진 모든 원소 a에 대해 $(a,a)\notin A$이면 비반사관계
☆ $(a,b)\in A$일 때, $(b,a)\in A$이면 대칭관계
☆ $(a,b)\in A$이고 $(b,a)\in A$일 때, $a=b$이면 반대칭관계
☆ $(a,b)\in A$이고 $(b,c)\in A$일 때, $(a,c)\in A$이면 추이관계

예제Ⅲ.3.6

관계의 성질은 집합에 관한 관계를 분류하는데 사용되며, 반사관계(reflexive relation), 대칭관계(symmetric), 추이관계(transitive)등으로 나누어 볼 수 있다. 여기서 추이관계란 모든 a, b, c ∈ A에 대하여 (a, b) ∈ R 이고 (b, c) ∈ R 이면 (a, c) ∈ R일 때 R은 추이관계(transitive)이다. 집합 A = {1, 2, 3}에 대하여 추이관계가 성립하지 <u>않는</u> 것은?(2006 교원)

① R = {(1,1), (1,2), (1,3), (2,1), (2,2), (2,3), (3,1), (3,2), (3,3)}
② R = {(2,2), (3,3)}　③ R = {(1,3), (2,2), (3,1)}
④ R = {(1,3)}　⑤ R = {(1,1), (1,2), (1,3), (2,3)}
<풀이> ③
추이관계가 성립하려면 (1,3)과 (3,1)이 있는 경우에는 반드시 (1,1)과 (3,3)이 있어야 한다.

문제Ⅲ.3.6
문제Ⅲ.3.2의 어떤 관계가 추이관계인가?
<풀이> ①, ②, ③, ④
R_{11}과 R_{12}에서 (1,3)과 (3,1)이 있을 경우 반드시 (1,1)과 (3,3)이 있어야 추이관계가 성립한다.

라. 동치관계

1) 동치관계

집합 A에 관한 관계 R이 반사, 대칭, 추이관계가 성립하면 관계 R을 동치관계(equivalence relation)라고 한다. 동치관계 R을 방향그래프로 표현해보면 모든 정점들은 루프를 가지고 있고 a에서 b로의 간선이 존재한다면 b에서 a로의 간선이 반드시 존재한다. 또한 추이관계이므로 정점 a에서 b로의 간선과 b에서 c로의 간선이 존재할 때 a에서 c로의 간선이 존재한다.

예제Ⅲ.3.7
집합 $A=\{1,2,3,4\}$ 에 대하여
$R=\{(1,1),(1,2),(2,1),(2,2),(3,3),(3,4),(4,3),(4,4)\}$ 이 동치관계임을 보이시오.
<풀이>
$(1,1),(2,2),(3,3),(4,4)$ 이 모두 존재하므로 반사관계이다. …①
$(1,2)$ 에 대해 $(2,1)$ 이 있고 $(3,4)$ 에 대해 $(4,3)$ 이 존재하므로 대칭관계이다. …②
$(1,2)$ 과 $(2,1)$ 에 대해 $(1,1),(2,2)$ 이 있고 $(3,4)$ 과 $(4,3)$ 에 대해 $(3,3),(4,4)$ 이 존재하므로 추이관계이다. …③
①, ②, ③에 의하여 R은 동치관계이다.

문제Ⅲ.3.7

정수의 집합인 Z에서의 관계 $R=\{(a,b)|a-b$가 5로 나누어진다$\}$가 동치관계인지를 보이시오.

<풀이>

$a-a=0=5\cdot 0$이므로 반사관계이다. …①

$a-b=5k\ (k\in Z)$이라면 $b-a=-a+b=-(a-b)=-5k=5\cdot(-k)$ 이므로 대칭관계이다. …②

$a-b=5m$, $b-c=5n\ (m,\ n\in Z)$이면, $a-c=a-b+b-c=5m+5n=5(m+n)$ 이므로 추이관계이다. …③

①, ②, ③에 의하여 R은 동치관계이다.

2) 동치류(Equivalence Class)

R이 집합 A에 관한 동치관계이면, R에 관한 a의 동치류를 $[a]$로 표기하고 $[a]=\{x|(a,x)\in R\}$로 정의한다. 하나의 집합을 중복되는 부분이 없도록 나누는 것을 집합의 분할(partition)이라고 한다. 즉, 분할은 집합을 구성하는 원소가 서로 소이고 각 분할의 합집합이 원래의 집합과 서로 같아야 한다. R에 의한 집합 A의 상집합(몫집합, quotient set)은 A의 동치류의 모임으로서 관계 R에 의한 집합 A의 분할이고 $A/R=\{[a]|a\in A\}$로 나타낸다.

예제Ⅲ.3.8

예제Ⅲ.3.7의 관계 R의 모든 동치류와 R에 의한 집합 A의 상집합을 구하여라.

<풀이> $A/R=\{[1],[3]\}=\{\{1,2\},\{3,4\}\}$

관계 R에서 $[1]=\{1,2\}$, $[2]=\{1,2\}$, $[3]=\{3,4\}$, $[4]=\{3,4\}$이다.

$[1]=[2]$ 이고 $[3]=[4]$ 이므로 $A/R=\{[1],[3]\}=\{\{1,2\},\{3,4\}\}$이다.

문제Ⅲ.3.8

문제Ⅲ.3.7의 관계 R의 모든 동치류를 구하여라.

<풀이> [0], [1], [2], [3], [4]

$[0]=\{\cdots,-10,\ -5,\ 0,\ 5,\ 10,\cdots\}$, $[1]=\{\cdots,\ -9,\ -4,\ 1,\ 6,\ 11,\ \cdots\}$,

$[2]=\{\cdots,\ -8,\ -3,\ 2,\ 7,\ 12\cdots\}$, $[3]=\{\cdots,\ -7,\ -2,\ 3,\ 8,\ 13,\ \cdots\}$,

$[4]=\{\cdots,\ -6,\ -1,\ 4,\ 9,\ 14,\ \cdots\}$이다.

마. 부분순서관계

1) 부분순서관계

집합 A에 관한 관계 R이 반사, 반대칭, 추이관계가 모두 성립하면 관계 R을 부분순서관계(반순서 관계, partial ordered relation)라고 한다. 이 때 (A,R)을 부분순서집합(partially ordered set, poset)이라 한다. 부분순서집합에 관한 관계를 나타내기 위해서 기호$\preccurlyeq$($\preceq$ 혹은 $\precsim$)를 사용한다. 정수의 집합 Z에 대하여 관계인 크거나 같다($\geqq$), 작거나 같다($\leqq$), 나누어 떨어진다(|)와 집합에서의 관계인 부분집합(⊂, ⊃)는 부분순서관계이다. 그러나 크다(>)나 작다(<)는 반사관계가 성립하지 않으므로 부분순서관계가 아니다.

부분순서집합 $(A,\preccurlyeq)$에서 집합 A의 두 원소 $a,\ b$가 $a\preccurlyeq b$이거나 $b\preccurlyeq a$이면 a와 b를 비교가능(comparable)하다고 하고 집합 A에 관한 관계 R에서 모든 원소가 비교가능하면 A를 선형순서집합(linearly ordered set) 혹은 전순서집합(totally ordered set)이라 하며, 이 때 부분순서관계 R을 선형순서관계(linearly ordered relation) 혹은 전순서관계(totally ordered relation)라고 한다.

예제Ⅲ.3.9

부분 순서 집합 $(Z,\leqq)$에서 작거나 같다($\leqq$)는 선형 순서 관계인가?

<풀이> 선형순서관계이다.

정수의 집합 Z의 임의의 두 원소 $a,\ b$에서 $a\leqq b$이거나 $b\leqq a$이다. 따라서 $\leqq$는 선형순서관계이다.

문제Ⅲ.3.9

부분 순서 집합 $(Z, |)$에서 $|$(나누어 떨어진다)는 선형순서관계인가?

<풀이> 선형순서관계가 아니다.

정수의 집합 Z의 임의의 두 원소 a, b에서 $a|b$가 아니면서 $b|a$가 아닌 수가 존재하므로 선형순서관계가 아니다.

2) 하세도표

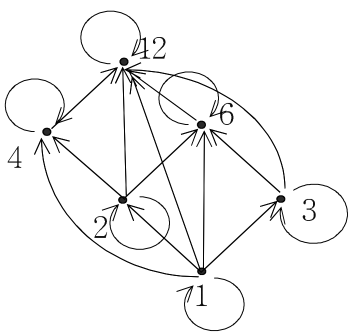
부분순서집합을 방향그래프로 그릴 때 기본적인 조건으로 반드시 있어야 할 간선들을 표시하지 않으면 보기가 훨씬 수월할 것이다. 모든 정점에서 루프를 가져야 하기 때문에 루프를 반드시 표시할 필요는 없다. 또한 추이관계를 나타내는 간선을 나타낼 필요가 없다. 마지막으로 모든 간선이 위로 향하고 있다고 가정하면 간선의 방향도 반드시 표시할 필요가 없어진다. 이렇게 부분순서집합의 방향그래프에서 반드시 존재해야 하는 간선을 제거하고 만든 도표를 하세도표 혹은 하세도형(hasse diagram, ordering diagram, poset diagram)이라고 한다. 하세도표를 그리는 순서는 다음과 같다.

① 원소를 정점으로 표시하고 반사관계를 나타내는 루프를 생략한다.
② 추이관계를 나타내는 간선을 제거한다.
③ 화살표를 생략하고 선으로 연결한다.

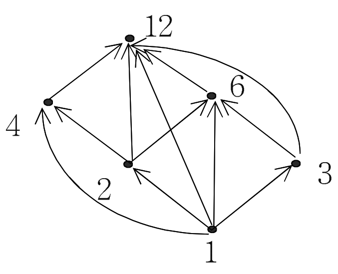
예제Ⅲ.3.10

집합 $A = \{1, 2, 3, 4, 6, 12\}$이고 부분순서관계 $|$(나누어 떨어진다)에 대한 하세도표를 그리시오.

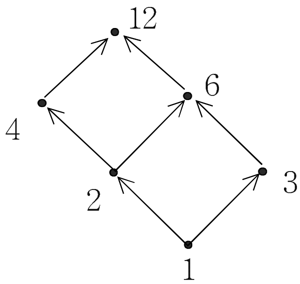
<풀이>



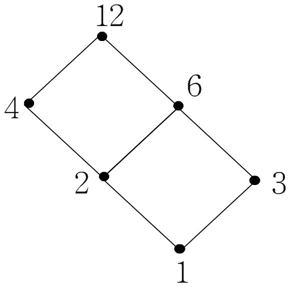
① 방향그래프



② 반사관계를 나타내는 화살표 생략



③ 추이관계를 나타내는 화살표 생략

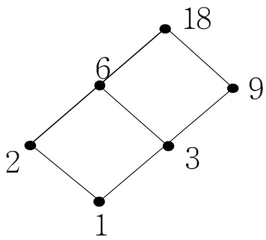


④ 화살표를 선으로 연결

문제Ⅲ.3.10

집합 $A=\{1, 2, 3, 6, 9, 18\}$이고 부분순서관계 $|$에 대한 하세 도표를 그리시오.

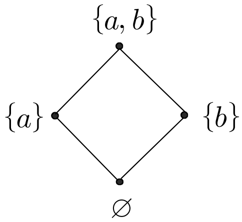
<풀이>



예제Ⅲ.3.11

부분순서집합$(P(\{a,b\}), \subset)$을 하세도표로 나타내시오.($(P(\{a,b\})$는 $\{a,b\}$의 모든 부분집합을 원소로 갖는 집합이다.)

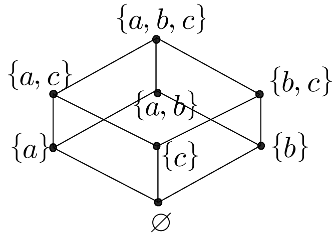
<풀이>



문제Ⅲ.3.11

부분순서집합 $(P(\{a,b,c\}), \subset)$을 하세도표로 나타내시오.

<풀이>



부분순서집합 $(A, \preccurlyeq)$이 있을 때 A의 원소 a에 대하여 원소 a가 다른 어떤 원소보다도 작지 않으면(즉, $a < b$인 원소 b가 A에 존재하지 않으면) a를 극대원소(maximal element)라고 한다. A의 원소 a에 대하여 원소 a가 어떤 원소보다도 크지 않으면(즉, $b < a$인 원소 b가 A에 존재하지 않으면) a를 극소원소(minimal element)라고 한다.

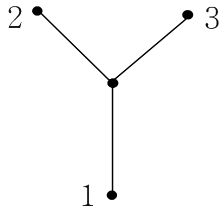
다른 어떤 원소보다도 큰 원소를 최대원소(greatest element)라고 하고 다른 어떤 원소보다도 작은 원소를 최소원소(least element)라고 한다.

$(A, \preccurlyeq)$는 부분순서집합이고 $a \in A$이고 $B \subset A$일 때, 모든 $x \in B$에 대하여 $x \preccurlyeq a$이면 a를 B에 대한 상한(upper bound)이라고 하고 모든 $x \in B$에 대하여 $a \preccurlyeq x$이면 a를 B에 대한 하한(lower bound)라고 한다. 상한 중에서 가장 작은 상한을 최소상한(least upper bound, LUB)이라고 하고 하한 중에서 가장 큰 하한을 최대하한(greatest lower bound, GLB)이라고 한다.

☆ a보다 더 큰 원소가 존재하지 않으면 a를 극대원소
☆ a보다 더 작은 원소가 존재하지 않으면 a를 극소원소
☆ 가장 큰 원소 a를 최대원소
☆ 가장 작은 원소 a를 최소원소
☆ 주어진 집합의 원소보다 크거나 같은 원소를 상한
☆ 주어진 집합의 원소보다 작거나 같은 원소를 하한
☆ 상한 중에서 가장 작은 상한을 최소상한
☆ 하한 중에서 가장 큰 하한을 최대하한

예제Ⅲ.3.12

다음 하세도표에서 극대원소, 극소원소, 최대원소와 최소원소를 찾아라.

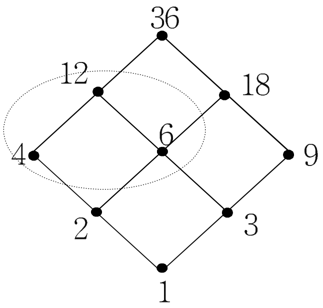


극대원소	2, 3
극소원소	1
최대원소	없다
최소원소	1

문제Ⅲ.3.12

임의의 정수 m의 약수인 집합을 D_m으로 표기할 때, D_{36}의 하세도표를 그리고 $\{4, 6, 12\} \subset D_{36}$의 상한, 하한, 최소상한, 최대하한을 구하여라.

<풀이>



상한	12, 36
하한	2, 1
최소상한	12
최대하한	2

연 습 문 제

1. 집합 A = {1, 2, 3}에 대하여 **추이관계가 성립하지 <u>않는</u> 것은?**

① R1 = ∅ ② R2 = {(1,2),(2,1)}
③ R3 = {(1,1),(2,2),(3,3)} ④ R4 = {(1,3)}
⑤ R5 = {(1,1), (1,2), (1,3), (2,1), (2,2), (2,3), (3,1), (3,2), (3,3)}

2. 집합 A={1,2,3}에 대한 다음 관계가 대칭관계 및 반대칭관계가 모두 성립하는 것은?

① R1 = {(1,3)} ② R2 = {(1,2),(2,1)}
③ R3 = {(1,1),(2,2),(3,3)} ④ R4 = {(1,1),(1,2),(2,1),(1,3)}
⑤ R5 = {(1,2),(1,3)}

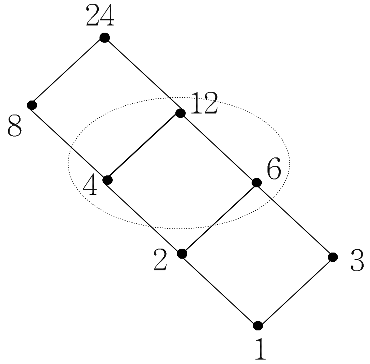
3. 집합 A={1,2,3}에서의 다음 관계 중에서 동치관계인 것은?

① R1 = {(1,1),(2,2)} ② R2 = {(1,2),(1,3)}
③ R3 = {(1,1), (1,2), (1,3), (2,2), (2,3), (3,3)} ④ R4 = ∅
⑤ R5 = {(1,1), (1,2), (1,3), (2,1), (2,2), (2,3), (3,1), (3,2), (3,3)}

4. 집합 A={1, 2, 3}에 대한 다음 관계가 부분순서관계인 것은?

① R1 = {(1,1),(2,2)}
② R2 = {(1,2),(1,3),(2,3)}
③ R3 = {(1,1), (1,2), (1,3), (2,1), (2,2), (2,3), (3,1), (3,2), (3,3)}
④ R4 = ∅
⑤ R5 = {(1,1), (1,2), (1,3), (2,2), (2,3), (3,3)}

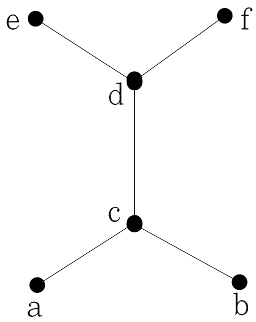
5 임의의 정수 m의 약수인 집합을 D_m으로 표기할 때, $D_{24} = \{1, 2, 3, 4, 6, 8, 12, 24\}$이고 이 부분순서집합의 하세도표는 다음과 같다. $A \subset D_{24}$인 $A = \{4, 6, 12\}$에 대해 다음 중 **바르지 못한 것은?**



① A의 극대원소와 최대원소는 12이다.
② A의 극소원소는 4와 6이다.
③ A의 최소원소는 없다.
④ A의 상한은 12와 24이다.
⑤ A의 최대하한은 1이다.

6 하세도표는 부분순서집합의 방향그래프에서 반드시 존재해야 하는 간선을 제거하고 만든 도표를 말한다. 다음 하세도표의 설명 중에서 바르지 못한 것은?

① a는 극대원소이다.
② b는 극소원소이다.
③ e는 극대원소이다.
④ f는 극대원소이다.
⑤ f는 최대원소이다.



풀 이

1 ②

R2 = {(1,2),(2,1)}에서 (1,2)와 (2,1)이 있을 때, 추이관계가 성립하려면 (1,1)과 (2,2)가 있어야 된다.

2 ③

①은 (1,3)만 있으므로 대칭관계가 성립하지 않는다.
②는 (1,2)과 (2,1)이 있으므로 반대칭관계가 성립하지 않는다.
④에는 (1,3)만 있으므로 대칭관계가 성립하지 않고 (1,2)와 (2,1)이 있으므로 반대칭관계가 성립하지 않는다.
⑤는 (1,2)과 (1,3)이 있으므로 대칭관계가 성립하지 않는다.

3 ④

①에는 (3,3)이 없으므로 반사관계가 아니다.
②는 (1,1), (2,2), (3,3)이 없으므로 반사관계가 성립하지 않고 (1,2)가 있는데 (2,1)이 없으므로 대칭관계도 성립하지 않는다.
③에는 (1,2)가 있으나 (2,1)이 없으므로 대칭관계가 성립하지 않는다.
⑤는 (1,1), (2,2), (3,3)이 없으므로 반사관계가 성립하지 않는다.

4 ⑤

①은 (3,3)이 없으므로 반사관계가 성립하지 않는다.
②는 (1,1), (2,2), (3,3)이 없으므로 반사관계가 성립하지 않는다.
③에는 (1,2)와 (2,1)가 있으므로 반대칭관계가 성립하지 않는다.
④는 (1,1), (2,2), (3,3)이 없으므로 반사관계가 성립하지 않는다.

5 ⑤

$A=\{4, 6, 12\}$의 극대원소, 극소원소, 최대원소, 최소원소, 상한, 하한, 최소상한, 최대하한을 구한 결과는 다음과 같다.

극대원소	12	상한	12, 24
극소원소	4, 6	하한	2, 1
최대원소	12	최소상한	12
최소원소	없다	최대하한	2

6 ⑤

e와 f는 극대원소이기는 하지만 최대원소는 아니다.

4. 그래프

그래프(graph) 이론은 18C 오일러에 의해 처음으로 연구되어 수학, 전산학, 공학 등과 같은 여러 가지 분야에 직면한 문제들을 해결하기 위해 이용되어 왔으며 현재 컴퓨터 과학 분야에 가장 많이 사용되는 수학적 모델이다.

그래프는 정점(꼭지점, 점, 결점, vertex, point, node)과 간선(선, 변, 연결선, edge, line, arc)으로 구성된 구조를 가진다. 일반적으로 정점은 이름을 가진 점 또는 원으로 표시되고 간선은 두 정점을 연결하는 선으로 나타낸다. 지하철 노선도는 역을 정점으로 역과 역 사이는 간선으로 나타낸 그래프라고 볼 수 있다. 그래프에서 정점의 위치나 간선의 길이는 의미가 없으며 간선의 연결 상태만이 중요하다.

가. 그래프

1) 그래프의 정의

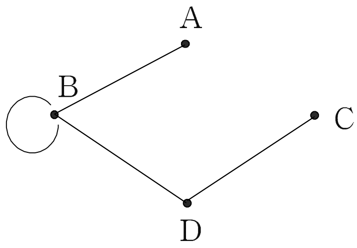
공집합이 아닌 유한개의 정점들의 집합인 $V=\{v_1, v_2, \cdots, v_m\}$ 와 간선들의 집합 $E=\{e_1, e_2, \cdots, e_n\}$로 이루어진 (V,E)를 그래프라고 하고 $G=(V,E)$로 나타낸다. 간선을 나타내는 정점의 쌍에 순서가 없도록 E의 각 원소를 집합 V의 두 원소쌍에 대응시키는 그래프를 무방향그래프(무향그래프, undirected graph)라고 하고 각 간선은 선으로 나타낸다. 또한, 집합 E의 각 원소를 집합 V의 두 원소의 순서쌍에 대응시키는 그래프를 방향그래프(유향그래프, directed graph, digraph)라고 하고 각 간선은 화살표를 사용하여 나타낸다.

이 책에서 무방향그래프는 그래프라고 줄여 쓰고 방향그래프와 구별하도록 하겠다.

예제Ⅲ.4.1

$V=\{A, B, C, D\}$, $E=\{(A,B),(B,B),(B,D),(C,D)\}$ 인 그래프 G를 그리시오.

<풀이>

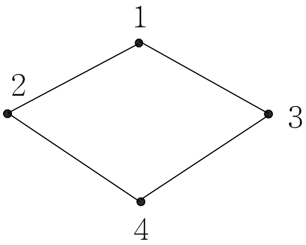


[그림Ⅲ.4.1] 그래프

문제Ⅲ.4.1

$V=\{1,2,3,4\}$, $E=\{(1,2),(1,3),(2,4),(3,4)\}$ 인 그래프 G를 그리시오.

<풀이>



정점 v_i와 정점 v_j를 연결한 간선 e_k가 존재할 때, 정점 v_i와 정점 v_j는 인접(이웃, adjacent, neighbor)한다고 하고, 간선 e_k는 정점 v_i와 정점 v_j에 접한다(교차, 접속, incident)라고 한다. v_i와 v_j를 간선 e_k의 끝점(end points)이라고 하며 간선의 끝점이 하나인 경우는 없고 간선은 반드시 정점으로 끝난다.

어떤 정점에 접하는 간선들의 수를 정점의 차수(degree)라고 한다. 루프(loop)란 정점으로부터 자기 자신으로 가는 간선이므로 하나의 루프는 그 정점의 차수를 2씩 증가시킨다. [그림Ⅲ.4.1]에서 정점 A의 차수는 1, 정점 B의 차수는 4, 정점 C의 차수는 1, 정점 D의 차수는 2이다. 한 간선은 반드시 두 정점에 접하므로 정점들의 차수의 총합은 간선 개수의 2배가 된다.

정점	A	B	C	D	계
차수	1	4	1	2	8

예제Ⅲ.4.2

10개의 정점과 각 정점에서의 차수가 6인 그래프의 간선 개수를 구하여라.

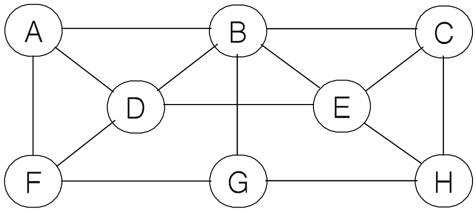
<풀이> 30

차수의 총합은 60이고 간선 개수의 2배이므로 간선의 개수는 30개이다.

문제Ⅲ.4.2

다음 그래프에서 분지수(degree)가 가장 큰 정점(vertex)의 분지수는?
(2003 초등)

① 3개
② 4개
③ 5개
④ 6개
⑤ 7개



<풀이> ③

각 정점의 차수는 다음과 같다.

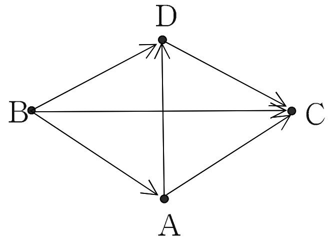
정점	A	B	C	D	E	F	G	H	계
차수	3	5	3	4	4	3	3	3	28

방향그래프에서 정점 v_i로 들어오는 간선의 개수를 정점 v_i의 진입차수(내차수, in-degree)라고 하고 정점 v_i에서 나가는 간선의 개수를 정점 v_i의 진출차수(외차수, out-degree)라고 하며 정점 v_i에서 진입차수와 진출차수의 합을 그 정점의 차수라고 하며 무방향그래프처럼 차수의 총합은 간선의 개수의 2배이다. 즉, 진입차수의 합과 진출차수의 합은 같고 간선의 개수와도 같다. 정점의 진입차수가 0인 경우에 그 정점을 원시(소스, source)라고 하고 정점의 진출차수가 0인 경우에는 그 정점을 종착(싱크, sink)이라고 한다.

예제Ⅲ.4.3

다음 방향그래프에서 진출차수가 가장 큰 정점과 진출차수를 구하시오.

① A, 3 ② B, 3 ③ C, 3 ④ B, 2 ⑤ D, 2



<풀이> ②

각 정점에 대한 진입차수와 진출차수는 다음과 같다.

정점	진입차수	진출차수
A	1	2
B	0	3
C	3	0
D	2	1
계	6	6

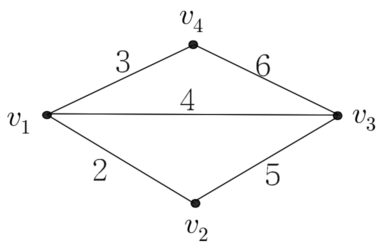
문제Ⅲ.4.3

진입차수의 합이 30인 방향그래프의 간선의 개수는?

① 10 ② 15 ③ 20 ④ 30 ⑤ 60

<풀이> ④ 방향그래프에서는 진입차수의 합과 진출차수의 합은 같고 간선의 개수와도 같다.

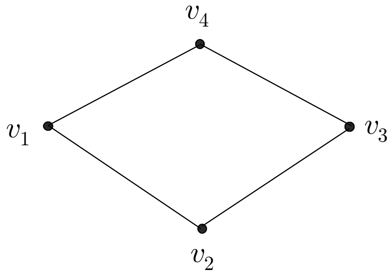
가중그래프(weighted graph)는 그래프의 간선에 가중치가 주어진 그래프를 말하고 가중치란 간선을 통과하는데 걸리는 시간이나 거리 등의 비용을 표현하는데 사용된다. [그림Ⅲ.4.2]는 가중그래프의 예이다.



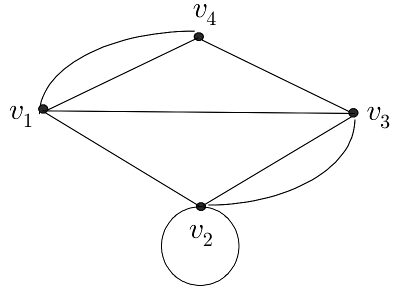
[그림Ⅲ.4.2] 가중그래프

한 정점에 인접한 정점이 없는 정점을 고립점(isolated vertex)이라 하며 고립점의 차수는 0이 된다. 루프가 없고 두 정점을 연결하는 간선이 하나뿐인 그래프를 단순그래프(simple graph)라고 하고 단순그래프가 아닌 것을 다중그래프(multi graph)라고 한다.

예를 들어, 〔그림Ⅲ.4.3〕은 단순그래프를, 〔그림Ⅲ.4.4〕는 다중그래프를 보여 준다.



[그림Ⅲ.4.3] 단순그래프



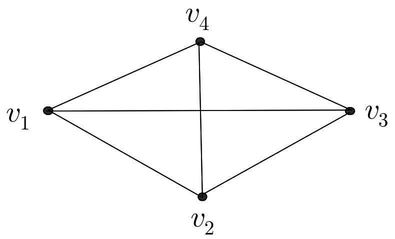
[그림Ⅲ.4.4] 다중그래프

그래프에서의 정점 v_0 에서 정점 v_n 까지의 경로(path)는 $\{v_0,v_1,\cdots,v_i,v_j,\cdots,v_{n-1},v_n\}=\{e_1,e_{2,}\ \cdots,e_k,\cdots,e_n\}$, $e_k=(v_{k-1},\ v_k)$ 로 나타내고 변의 수를 경로의 길이(length)라 한다. 모든 정점이 연결되어 있는 그래프를 연결그래프(connected graph)라고 한다. 경로가 같은 간선을 두 번 포함하지 않는 경로를 단순경로(simple path)라 하며 어떤 정점도 두 번 지나지 않는 경로를 기본경로(elementary path)라고 한다.

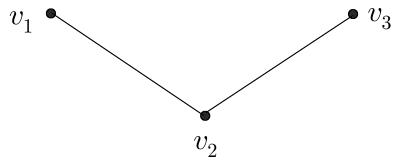
	정점 중복 허용	간선 중복 허용
경　　로	○	○
단순경로	○	×
기본경로	×	×

경로$(v_0, v_1, v_2, \cdots, v_i, v_j, \cdots, v_n)$ 에서 $v_i = v_j$ 인 경로를 순환(cycle)라 하며 어떤 간선을 반복하여 지나가지 않는 순환을 단순순환(simple cycle)라고 하며 v_0 를 제외한 어떠한 정점도 반복하여 지나가지 않는 순환을 기본순환(elementary cycle)이라 한다.

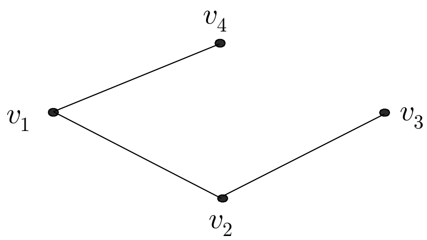
$V_1 \subset V$이고 $E_1 \subset E$인 $G_1 = (V_1, E_1)$을 $G=(V,E)$의 부분그래프(subgraph)라 한다. $V_2 = V$이고 $E_2 \subset E$인 $G_2 = (V_2, E_2)$를 $G=(V,E)$의 신장부분그래프(spanning subgraph)라 한다. [그림Ⅲ.4.6]은 [그림Ⅲ.4.5]에 있는 그래프 G의 부분그래프이고 [그림Ⅲ.4.7]는 [그림Ⅲ.4.5]의 그래프 G의 신장부분그래프의 예이다.



[그림Ⅲ.4.5] 그래프 G



[그림Ⅲ.4.6] 부분그래프

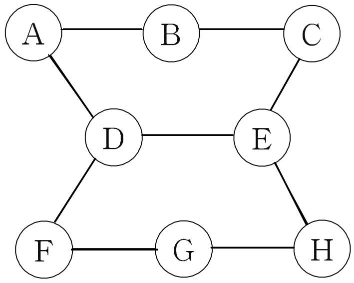


[그림Ⅲ.4.7] 신장부분그래프

예제Ⅲ.4.4

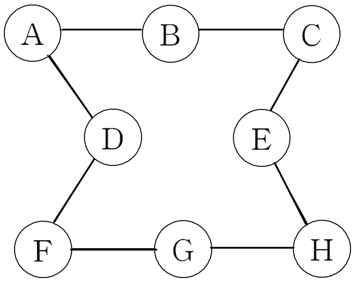
다음과 같은 그래프에서 가장 긴 사이클(cycle)은 몇 개의 간선(edge)을 포함하는가? (2002초등)

① 3개
② 4개
③ 5개
④ 7개
⑤ 8개



<풀이> ⑤

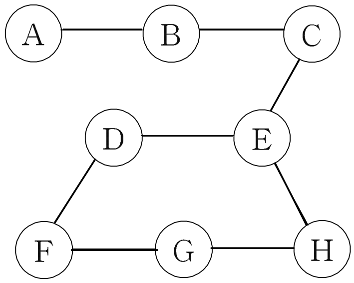
가장 긴 사이클은 A, B, C, E, H, G, F, D, A이므로 포함된 간선의 수는 8개이다.



문제Ⅲ.4.4

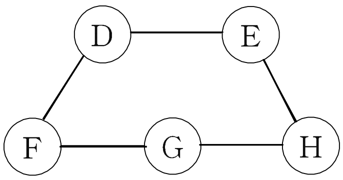
다음과 같은 그래프에서 가장 긴 사이클(cycle)은 몇 개의 간선(edge)을 포함하는가?

① 3개
② 4개
③ 5개
④ 7개
⑤ 8개



<풀이> ③

가장 긴 사이클은 D, E, H, G, F, D이므로 포함된 간선의 수는 5개이다.



2) 그래프의 표현방법

그래프는 그림으로 표현하는 것이 가장 이해하기는 쉽지만 컴퓨터에 표현하고 여러 가지 연산을 수행하기에는 많은 어려움이 있다. 그래서 컴퓨터에 그래프를 저장하기 위해서 인접행렬(adjacency matrix), 결합행렬(incidence matrix)과 인접리스트(adjacency list) 등을 사용한다. 어떤 한 방법이 다른 방법보다 좋다고 할 수는 없으며 주어진 문제와 시스템에 따라 최선의 방법을 결정해야 한다.

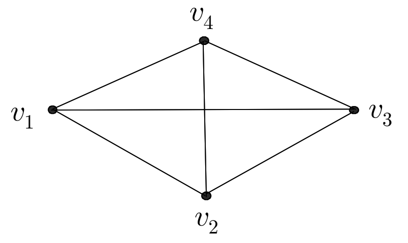
가) 인접행렬(adjacency matrices)

먼저 단순그래프 $G=(V,E)$에서 각 정점 사이의 간선의 존재여부를 나타내는 행렬인 G의 인접행렬은 정점들을 나열한 것으로 $n \times n$행렬이다. 인접행렬을 $A=(a_{ij})$이라 할 때 a_{ij}는 두 정점이 인접할 경우 1이고 두 정점이 인접하지 않을 경우 0의 값을 가진다. 즉,

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

이다.

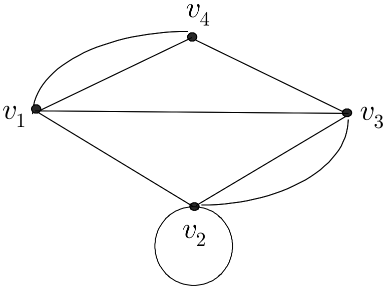
인접행렬은 직관적이고 쉽게 그래프를 표현할 수 있다는 장점이 있으나 항의 대부분이 0인 희소행렬인 경우 기억 공간을 낭비하는 경향이 있다. 단순그래프의 인접행렬은 대칭행렬, 즉, $a_{ij}=a_{ji}$이다. 단순그래프는 루프를 갖지 않기 때문에 $a_{ii}=0$이 된다. 그래프를 인접행렬로 표현한 예는 아래와 같다.



$$\begin{bmatrix}0&1&1&1\\1&0&1&1\\1&1&0&1\\1&1&1&0\end{bmatrix}$$

인접행렬은 다중그래프나 방향그래프를 표현하는데도 사용될 수 있다. 먼저 다중그래프에서 루프가 있으면 a_{ii}가 1이 되고 다중 간선이 존재하면 a_{ij}는 v_i와 v_j사이의 간선의 수와 같다. 따라서 다중그래프도 대칭행렬이다.

다중그래프를 인접행렬로 표현한 예는 아래와 같다.



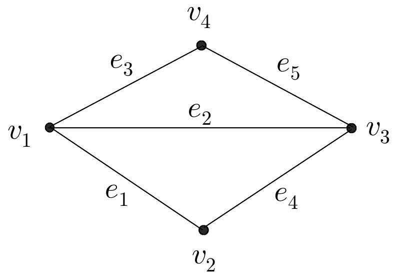
$$\longrightarrow \begin{bmatrix} 0 & 1 & 1 & 2 \\ 1 & 1 & 2 & 0 \\ 1 & 2 & 0 & 1 \\ 2 & 0 & 1 & 0 \end{bmatrix}$$

나) 결합행렬(교차행렬, incidence matrix)

결합행렬은 각 정점과 간선의 연결 상태를 나타낸다. 결합행렬의 행은 정점이고 열은 간선이 되는 $m\times n$행렬이다. 결합행렬을 $M=(m_{ij})$이라 할 때 간선이 정점에 연결된 경우에는 m_{ij}는 1이고 연결되지 않은 경우에는 m_{ij}는 0이다.
즉,

$$m_{ij}=\begin{cases} 1, & \text{간선 } e_k\text{가 정점 } v_i\text{에 연결된 경우} \\ 0, & \text{간선 } e_k\text{가 정점 } v_i\text{에 연결되지 않은 경우} \end{cases}$$

〔그림Ⅲ.4.8〕의 그래프 G를 결합행렬 A로 표현하면 다음과 같다.



[그림Ⅲ.4.8] 그래프 G

$$A=\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

다) 인접리스트

인접행렬이 희소행렬이 되면 기억장소가 낭비되므로 이런 경우 인접행렬의 단점을 보완하여 저장 장소의 효율성을 높이기 위한 표현 방법이 인접리스트이다. 인접리스트의 각 정점은 정점과 포인터라는 두 개의 필드가 있고 **각 정점에 연결된 정점들을 연결리스트로 나타내는 것으로** 순서는 관계가 없다. 〔그림Ⅲ.4.9〕는 〔그림Ⅲ.4.8〕의 그래프 G를 인접리스트로 표현한 것이다.

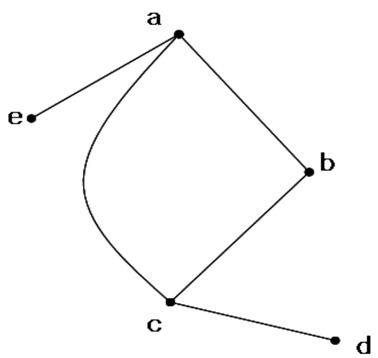
v_1 → v_2 → v_3 → v_4 nu
v_2 → v_1 → v_3 nu
v_3 → v_1 → v_2 → v_4 nu
V_4 → v_1 → v_3 nu

[그림Ⅲ.4.9] 인접리스트

예제Ⅲ.4.5

인접리스트(adjacency list)에서는 인접행렬의 n행들의 값을 n개의 연결리스트로 나타낸다. 인접리스트 방식은 각 정점에 인접한 정점들을 순서와 상관없이 일일이 열거하는 것이다. 다음 그래프의 인접리스트를 <u>잘못</u> 나타낸 것은 무엇인가?

① a → b → c → e → null
② b → a → c → null
③ c → a → b → e → null
④ d → c → null
⑤ e → a → null



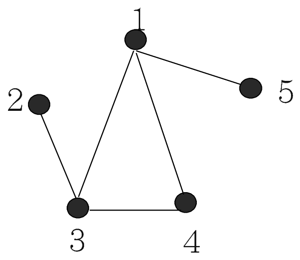
<풀이> ③

c에 인접한 정점은 a, b, d이므로 c → a → b → d → null이라고 나타내어야 한다.

문제Ⅲ.4.5

인접리스트(adjacency list)에서는 인접행렬의 n행들의 값을 n개의 연결리스트로 나타낸다. 인접리스트 방식은 각 정점에 인접한 정점들을 순서와 상관없이 일일이 열거하는 것이다. 다음 그래프의 인접리스트를 **잘못** 나타낸 것은 무엇인가?

① 1 → 3 → 4 → 5 → null
② 2 → 3 → null
③ 3 → 2 → 4 → null
④ 4 → 1 → 3 → null
⑤ 5 → 1 → null



<풀이> ③ 3에 인접한 정점은 1, 2, 4이므로 3 → 1 → 2 → 4→ null이라고 나타내어야 한다.

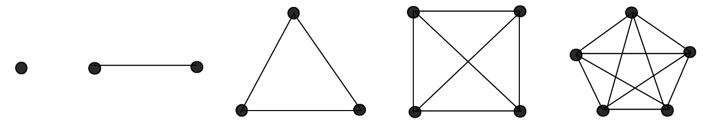
나. 그래프의 종류

1) 완전그래프(complete graph)

그래프에서 각 정점이 다른 모든 정점들과 연결되어 있는 그래프를 n개의 정점을 가진 완전그래프라고 하고 K_n으로 나타낸다. 완전그래프 K_n의 간선의 개수는 $\frac{n(n-1)}{2}$이다.

예제Ⅲ.4.6

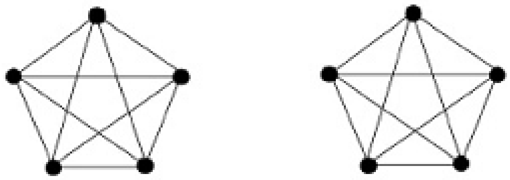
완전그래프 K_1, K_2, K_3, K_4, K_5를 그리고 간선의 개수를 구하시오.

완전그래프	K_1	K_2	K_3	K_4	K_5	K_n
						
간선의 개수	0	1	3	6	10	$\frac{n(n-1)}{2}$

문제Ⅲ.4.6

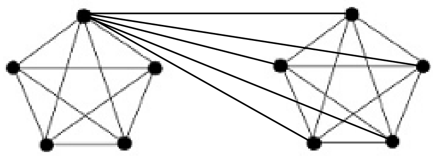
모든 정점(vertex) 사이에 간선(edge)이 존재하는 무방향그래프(undirected graph)를 완전그래프(complete graph)라고 한다. 정점의 개수가 5개인 두 개의 완전그래프를 하나로 이어서 정점의 개수가 10개인 완전그래프를 만들려고 한다. 몇 개의 간선을 추가로 넣어 주어야 하는가?

① 15개
② 18개
③ 20개
④ 22개
⑤ 25개



<풀이> ⑤

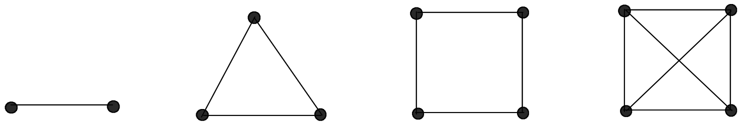
왼쪽과 오른쪽의 그래프가 따로 있다면 완전그래프이므로 왼쪽 그래프의 다섯 개의 각 정점에서 오른쪽 그래프의 다섯 개의 각 정점으로 간선을 그어주면 전체가 하나의 완전그래프가 된다. 따라서 5×5=25개의 간선을 추가로 넣어 주어야 한다.



〈별해〉 $K_{10} = \frac{10\times9}{2} = 45$ 이어야 하는데, 두 개의 K_5에는 각 10개씩의 간선이 있으므로 $45-20=25$개의 간선이 더 필요하다.

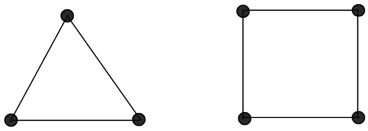
2) 정규 그래프(regular graph)

모든 정점의 차수가 같은 그래프를 정규 그래프라고 한다. 특히, 그래프 G의 모든 정점 v_i에 대하여 차수가 r인 경우 G를 r차 정규 그래프라고 하며 간선의 개수는 $\frac{1}{2}nr$이 된다. 완전그래프 K_n은 $n-1$차 정규 그래프가 된다. 특히 2차 정규 그래프를 순회그래프(circuit graph)라고 하고 정점이 n개인 순회 그래프를 C_n으로 나타낸다. [그림Ⅲ.4.10]은 정규그래프의 예이고 [그림Ⅲ.4.11]은 순회그래프의 예이다.



1차 정규그래프 2차 정규 그래프 3차 정규그래프

[그림Ⅲ.4.10] 정규그래프



C_3 C_4

[그림Ⅲ.4.11] 순회그래프

3) 이분그래프

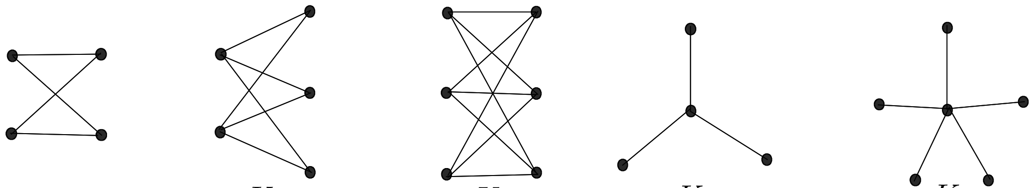
그래프 $G=(V,E)$에서 V가 두 부분집합 V_1과 $V_2=V-V_1$로 분할되어 각 간선이 V_1의 정점과 V_2의 정점의 쌍으로 나타내어지면 이분그래프(bipartite graph)라고 한다.

V_1의 모든 정점과 V_2의 모든 정점 사이에 간선이 존재하는 그래프를 완전 이분그래프(complete bipartite graph)라고 한다. V_1의 정점의 개수가 m개이고 V_2의 정점의 개수가 n개인 완전이분그래프는 $K_{m,n}$으로 나타내고 $K_{m,n}$의 간선의 개수는 $m\times n$개다. 또한 $K_{1,m}$을 성그래프(star graph)라고 하며 간선의 개수는 m개다.

예제Ⅲ.4.7

완전이분그래프 $K_{2,2}$, $K_{2,3}$과 $K_{3,3}$와 성 그래프 $K_{1,3}$와 $K_{1,5}$를 그리시오.

<풀이>



$K_{2,2}$ $K_{2,3}$ $K_{3,3}$ $K_{1,3}$ $K_{1,5}$

문제Ⅲ.4.7

완전이분그래프 $K_{2,2}$, $K_{2,3}$과 $K_{3,3}$의 간선의 개수를 구하시오.

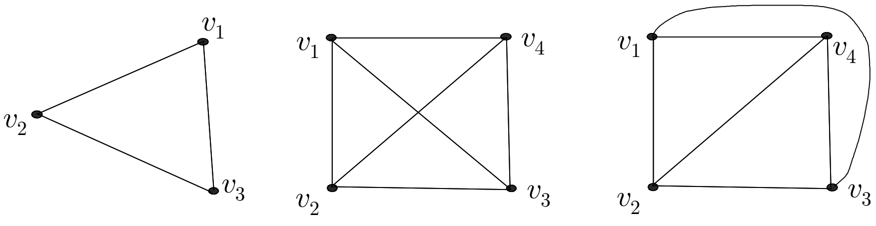
<풀이>

완전이분그래프	$K_{2,2}$	$K_{2,3}$	$K_{3,3}$	$K_{m,\,n}$
간선의 개수	4	6	9	mn

4) 평면그래프

그래프의 어떠한 두 간선도 정점이외에서는 서로 만나지 않고 평면상에 그려지는 그래프를 평면그래프(planar graph)라고 불린다. [그림Ⅲ.4.12]는 평면그래프의 예이다. 간선이 교차한 경우도 교차하지 않도록 다시 그릴 수 있다면 평면그래프가 된다. [그림Ⅲ.4.12]의 (2)번 그래프는 간선이 교차하지만 정점 v_1에서 v_3으로 가는 간선을 다시 그리면 (3)번과 같은 평면그래프가 된다.

(1) (2) (3)

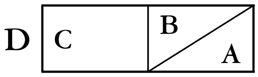


[그림Ⅲ.4.12] 평면그래프

평면그래프를 평면상에 그려서 간선을 따라 평면을 가위로 자르면 평면은 여러 개의 조각으로 분할되는데, 이 조각을 영역(region)이라고 한다. 영역의 차수는 영역을 둘러싸고 있는 간선들의 순환의 길이이며, 평면그래프에서 영역들의 차수의 총합은 간선의 개수의 두 배이다. 평면그래프에서 $|V|$를 정점의 수, $|E|$를 간선의 수, $|R|$을 영역의 수라고 하면 $|V|-|E|+|R|=2$이다. K_5와 $K_{3,3}$은 평면그래프가 아니며 K_5나 $K_{3,3}$을 부분 그래프로 포함하고 있는 그래프는 평면그래프가 될 수 없다.

예제Ⅲ.4.8

연결된 그래프는 그래프안의 임의의 정점 v에서 w로의 경로가 존재하는 그래프이고, 평면그래프는 간선을 겹치지 않게 평면에 그릴 수 있는 그래프를 의미한다. 연결된 평면그래프가 평면에 그려지면 그 평면은 면(face)이라고 하고 연속된 영역으로 나누어진다. 예를 들어, 연결된 평면그래프



는 4개의 면을 갖는다. 연결된 평면그래프에서 면의 개수는 정점과 간선의 수에 의해 구할 수 있는데, 연결된 평면그래프가 9개의 정점을 가지고 있고, 각 정점의 차수가 2, 2, 2, 3, 3, 3, 4, 4, 5일 때 간선의 개수와 면의 개수는 얼마인가?(2004 교원)

① 간선 : 13개 면 : 6개
② 간선 : 13개 면 : 7개
③ 간선 : 14개 면 : 6개
④ 간선 : 14개 면 : 7개
⑤ 간선 : 15개 면 : 8개

<풀이> ④

정점의 수 $v=9$, 간선의 수는 차수의 총합의 $\frac{1}{2}$이므로 간선의 수 $e=14$, $v-e+f=2$이므로 면의 수 $f=7$이다.

문제Ⅲ.4.8

그래프 G를 평면에 그릴 때 어떠한 간선도 정점이 아닌 곳에서는 교차하지

않도록 그릴 수 있다면 G를 평면그래프라고 한다. 그래프 G를 연결된 평면그래프라고 하고 정점의 수를 v, 간선의 개수를 e, G에 의해 평면상에 형성되는 영역의 수를 r 이라고 하면, 다음 중에서 오일러의 공식을 올바르게 나타낸 것은?(2007 교원)

① $r=e-v+1$ ② $r=e-v+2$ ③ $r=v-e+1$
④ $r=v-e+2$ ⑤ $r=e-v-1$

<풀이> ②
$v-e+r=2$이므로 $r=e-v+2$이다.

5) 오일러그래프

오일러그래프는 독일의 쾨니히스베르그(Königsberg)의 프뢰겔(pregel)강을 지나는 7개의 다리를 단 한번만에 지나갈 수 있느냐의 문제에 대해 오일러가 이 문제를 단순화시켜 지날 수 없다고 해결한 것에서 유래한다.

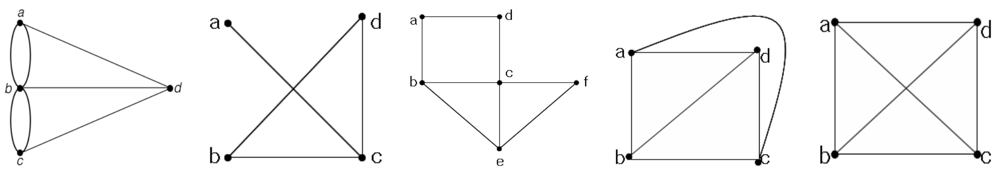
오일러경로(Euler path)란 모든 간선을 단 한번씩만 지나가는 경로이다. 또한 각 간선을 단 한번만 지나가는 순회를 오일러순회(Euler circuit)라고 하며, 오일러 순회를 갖는 그래프를 오일러그래프(Euler graph)라고 한다.

정점에 연결된 간선의 수가 홀수 개인 정점을 홀수점이라고 하고 연결된 간선의 개수가 짝수 개인 정점을 짝수점이라고 한다. 그래프에서 홀수점은 반드시 쌍으로 존재한다. 2개의 홀수점을 갖는 연결그래프는 오일러경로가 된다. 또한, 연결그래프의 어떠한 정점도 홀수점이 아닐 경우(즉, 모든 정점의 차수가 짝수일 때) 오일러순회가 된다.

예제Ⅲ.4.9

오일러는 쾨니히스베르크 프레겔강에 연결되어 있던 다리 일곱 개를 한 번씩만 지나 산책하면서 되돌아오는 경로가 존재하는지 여부를 밝히는데 그래프 이론을 사용하였다. 그래프 G=(V, E)에 대해 G안의 모든 정점(V)과 모든 간선(에지, E)을 포함되는 경로를 오일러경로라 한다. **오일러 경로가 존재하는 것으로 짝지어진 것은? (2006 교원)**

가 나 다 라 마



① 가 - 나　　② 가 - 라　　③ 나 - 다
④ 다 - 마　　⑤ 라 - 마

<풀이> ③

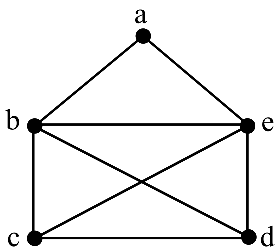
연결그래프에서 홀수점이 하나도 없거나 2개일 때 오일러경로가 된다.

그래프	가	나	다	라	마
홀수점의 개수	4	2	2	4	4

문제Ⅲ.4.9

어떤 도형을 펜을 떼지 않고 그리는 것을 한붓그리기라고 한다. 다음과 같은 도형을 한붓그리기로 그리려 한다. 시작점이 될 수 있는 점은 몇 개인가? (2005 초등)

① 0개
② 1개
③ 2개
④ 3개
⑤ 5개



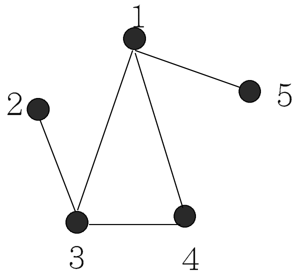
<풀이> ③

시작점이 될 수 있는 점은 홀수점인 c와 d이다.

예제Ⅲ.4.10

펜을 떼지 않고 도형을 그리는데 한 번 지나간 선을 다시 지나지 않고 그릴 수 있다면 그 도형은 한붓그리기가 가능하다고 한다. 물론 이 때 시작점으로 다시 돌아올 필요는 없다. 다음과 같은 도형이 한붓그리기가 가능하게 되기 위해서는 최소 몇 개의 선분을 추가해야 하는가? (2006 초등)

① 0개
② 1개
③ 2개
④ 3개
⑤ 4개



<풀이> ②

주어진 그래프에서 홀수점의 개수는 4개이다. 시작점으로 돌아올 필요가 없는 한붓그리기이므로 이 경우에 오일러 경로가 존재하게 하려면 현재 홀수점의 개수가 2이하가 되어야 한다. 그래서 추가해야 할 최소 선분의 개수는 1개이다.

문제Ⅲ.4.10

다음 중 한붓그리기가 가능한 것을 모두 나열한 것은?(2007 초중등)

A. 정점의 수가 10개인 완전그래프(complete graph)
B. 정점의 수가 9개인 완전그래프(complete graph)
C. 정점의 수가 15개인 완전이진트리(complete binary tree)

① A ② B ③ C ④ A, B ⑤ A, C

<풀이> ②

A. 정점의 수가 10개인 완전그래프는 각 정점에 연결된 간선의 수가 9개이므로 모두 홀수점이 되므로 한붓그리기를 할 수 없다.
B. 정점의 수가 9개인 완전그래프에서 각 정점에 연결된 간선의 수는 8개이므로 모두 짝수점이 된다. 따라서 어느 정점에서 시작하더라도 한붓그리기가 된다.
C. 정점의 수가 15개인 완전이진트리는 깊이가 4이고 14개의 홀수점을 가지므로 한붓그리기를 할 수 없다. 높이가 3이상인 완전이진트리부터는 한붓그리기를 할 수 없다.

예제Ⅲ.4.11

그래프 G=(V, E)에 대해 G 안의 모든 정점(V)과 모든 에지(E)가 포함되는 경로를 오일러 경로라 하고, 순환을 오일러 순환이라고 한다. 그래프 G=(V, E)가 오일러 순환을 갖기 위한 필요충분조건은 무엇인가?

① 그래프 G가 연결그래프이고 모든 정점들이 짝수 차수일 때다.
② 그래프 G가 연결그래프이고 1개의 정점이 홀수 차수일 때다.
③ 그래프 G가 연결그래프이고 2개의 정점이 홀수 차수일 때다.
④ 그래프 G가 연결그래프이고 3개의 정점이 홀수 차수일 때다.
⑤ 그래프 G가 연결그래프이고 4개의 정점이 홀수 차수일 때다.

<풀이> ①
오일러그래프가 되려면 연결그래프이면서 모든 정점들이 짝수점이 되어야 한다.

문제Ⅲ.4.11

각 정점의 차수는 4, 3, 3, 3, 3인 다섯 개의 정점을 가진 그래프가 있다. 이 그래프가 오일러 그래프가 되려면 최소한으로 몇 개의 간선을 추가해야 하는가?
① 1개 ② 2개 ③ 3개 ④ 4개 ⑤ 5개

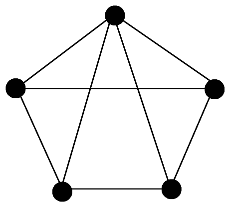
<풀이>
홀수점의 개수가 4개이므로 간선 두 개만 추가하면 된다.

6) 해밀턴그래프

하나의 그래프에서 각 정점을 단 한번만 지나가는 경로나 순회를 찾아내는 문제가 해밀턴에 의해 제기되었다. 해밀턴경로(Hamilton path)는 그래프에서 각 정점을 단 한번만 지나가는 경로를 말하고 해밀턴순회(Hamilton circuit)는 그래프의 각 정점을 단 한번만 지나가는 순회이다. 해밀턴 순회를 가지는 그래프를 해밀턴그래프(Hamilton graph)라고 한다. 해밀턴그래프를 찾는 방법은 아직 존재하지 않는다.

예제Ⅲ.4.12

다음과 같은 그래프에 대한 설명으로 옳지 않은 것은? (2004 초중등)



① 무방향(undirected) 그래프이다.
② 5개의 정점(vertex)과 8개의 간선(edge)로 구성되어 있다.
③ 분지수(degree)가 가장 큰 정점의 분지수는 4이다.
④ 모든 정점을 한 번씩만 지나서 시작점으로 돌아올 수 있다.
⑤ 모든 간선을 한 번씩만 지나서 시작점으로 돌아올 수 있다.

<풀이> ⑤

⑤에서 오일러 순회가 가능하려면 모든 정점이 짝수점이어야 하지만 주어진 그래프에는 2개의 홀수점이 있으므로 시작점으로 돌아올 수 없다.

문제Ⅲ.4.12

예제Ⅲ.4.9의 그래프에서 해밀턴그래프를 찾아보시오.

<풀이>

(가), (다), (라)와 (마)가 해밀턴그래프이다.

다. 그래프 탐색

그래프로 표현된 자료를 가지고 어떤 작업을 하기 위해서는 기본적으로 각 정점이나 간선을 체계적으로 탐색하는 방법이 필요하다. 그래프 탐색은 그래프의 모든 정점과 간선을 체계적으로 방문하는 것을 말하며 깊이우선탐색(depth first search, DFS)과 너비우선탐색(breadth first search, BFS)이 있다.

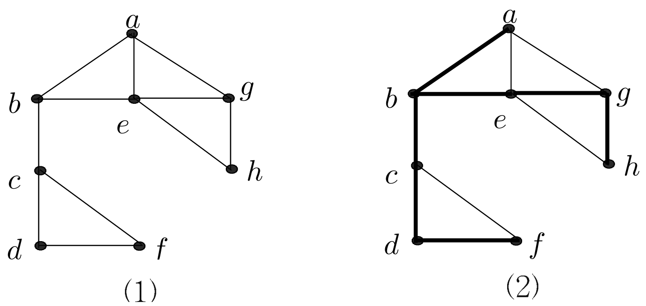
1) 깊이우선탐색

정점 v에 인접한 정점 중에서 아직 방문하지 않은 정점 w를 선택하여 그 w

에 대해 깊이우선탐색을 수행한다. 각 정점 v를 방문할 때 다음에 v에 인접한 다른 정점을 쉽게 찾기 위해서는 v의 위치를 스택(stack)에 저장해야 한다.

〔그림Ⅲ.4.13〕의 (1) 그래프에서 정점 a를 처음으로 탐색을 시작한다고 하자. a에 인접한 여러 정점 중에서 b를 임의로 선택하여 방문한다. 두 번째에 대한 규칙은 없지만, 동일 순서일 경우 보통 알파벳 순서가 빠르거나 첨자가 작은 것을 우선 방문한다. 다음에는 b에 인접한 정점들 중에서 c를 선택하여 방문한다. 이러한 식으로 d, f를 차례대로 방문하게 되면 f에서 더 이상 방문할 정점이 없게 되는데 그러면 f이전에 방문한 정점인 c로 돌아간 후, c에서 방문할 새로운 정점을 찾게 된다.

〔그림Ⅲ.4.13〕의 (2)의 굵은 선은 깊이우선탐색을 이용한 방문한 경로를 표시한 것이고 깊이우선탐색 결과는 a, b, c, d, f, e, g, h이다.



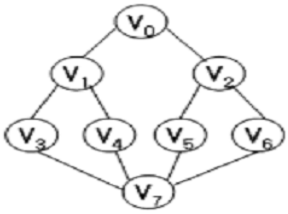
[그림Ⅲ.4.13] 깊이우선탐색

예제Ⅲ.4.13

다음 그래프의 정점 v_0에 깊이우선탐색(Depth-First Search)을 할 때, 마지막에 방문하는 정점은?(단, 조건이 같으면 첨자가 작은 정점을 먼저 방문한다.) (2006 교원)

① v_7 ② v_4 ③ v_2

④ v_5 ⑤ v_6

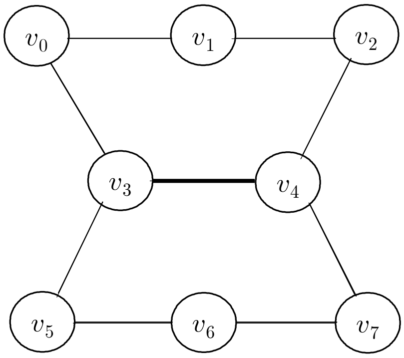


<풀이> ⑤

깊이우선탐색의 결과는 v_0 , v_1 , v_3 , v_7 , v_4 , v_5 , v_2 , v_6 이다.

문제Ⅲ.4.13

다음 그래프의 정점 v_0에 깊이우선탐색(Depth-First Search)을 할 때, 마지막에 방문하는 정점은?(단, 조건이 같으면 첨자가 작은 정점을 먼저 방문한다.)



① v_7　② v_4　③ v_2
④ v_5　⑤ v_6

<풀이> ①

깊이우선탐색의 결과는 v_0 , v_1 , v_2 , v_4 , v_3 , v_5 , v_6 , v_7 이다.

2) 너비우선탐색

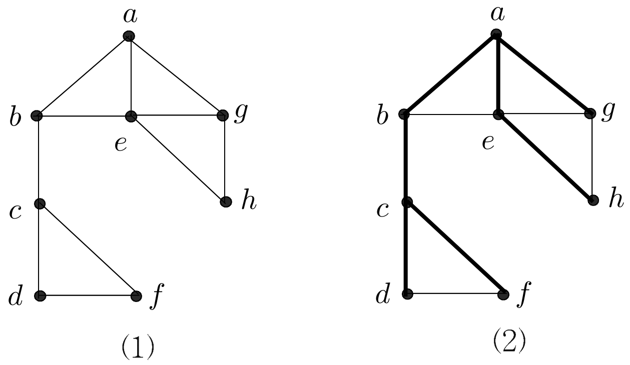
너비우선탐색은 현재 방문 중인 노드의 인접한 정점들을 우선적으로 모두 방문한 후에만 다음 정점을 방문할 수 있다. 방문되는 점들을 큐(queue)에 저장한다. 큐에서 한 정점을 꺼내 한 정점에 인접한 모든 정점들을 조사한 후 방문하지 않은 정점들을 방문한 후 큐에 넣는다. 이미 방문된 정점들은 무시하고 너비우선탐색을 수행한다.

〔그림Ⅲ.4.14〕의 (1) 그래프에서 정점 a를 방문하는 것으로 탐색을 시작한다고 하자. a의 인접한 여러 정점 b, e, g를 우선적으로 모두 방문한다. 이제 a의 인접한 노드는 다 방문하였기 때문에 방금 방문한 정점 중에서 b를 임의로 선택한 후,(깊이우선 탐색처럼 편의상 알파벳 순서가 빠른 것을 우선 선택하는 것이 유리하다.) b의 인접한 정점을 모두 찾는다. 여기서는 c를 방문한다.

다음에는 a의 인접한 정점 중에서 b의 다음으로 방문한 e에 인접한 정점 중에서 아직까지 방문하지 않은 정점인 h를 방문한다. 위의 과정을 계속하면

모든 정점들을 방문하게 된다.

〔그림Ⅲ.4.14〕의 (2)의 굵은 선은 너비우선탐색을 이용한 방문한 경로를 표시한 것이고 너비우선탐색 결과는 a, b, e, g, c, h, d, f 이다.

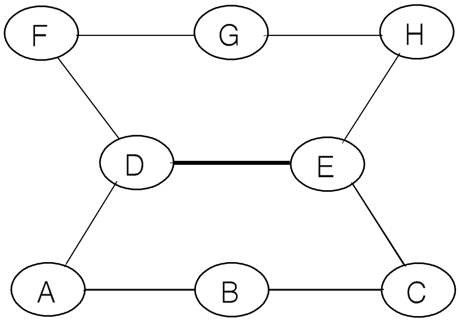


[그림Ⅲ.4.14] 너비우선탐색

예제Ⅲ.4.14

다음과 같은 그래프에서 정점(vertex) A부터 너비우선탐색(Breadth First Search, BFS)을 할 때 보기 중 정점들의 방문 순서가 될 수 없는 것은? (단, 조건이 같으면 알파벳 순서가 빠른 것을 우선 방문한다.)(2002 고등)

① A-B-D-C-E-F-H-G
② A-B-D-C-E-H-F-G
③ A-B-D-C-F-E-G-H
④ A-D-B-E-F-C-H-G
⑤ A-D-B-F-E-C-G-H



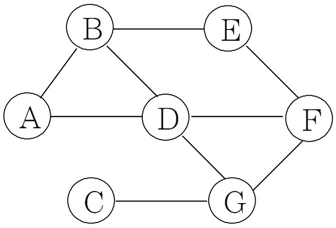
<풀이> ②

너비우선탐색에서 ②번처럼 먼저 방문한 노드의 하위노드를 우선 방문해야 한다. 그래서 E와 F사이에 H를 우선 방문할 수 없다.

문제Ⅲ.4.14

다음과 같은 정점 A부터 시작하여 너비우선탐색(BFS)으로 탐색하려고 한다. 가장 늦게 방문하게 되는 정점은?(단, 조건이 같으면 알파벳 순서가 빠른 것을 우선 방문한다.)(2005 중고등)

① C
② D
③ E
④ F
⑤ G



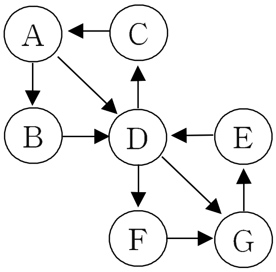
<풀이> ①

너비우선탐색의 결과는 A, B, D, E, F, G, C이다.

예제Ⅲ.4.15

다음과 같은 방향그래프의 정점 A에서 너비우선탐색(BFS, breadth-first search)을 할 때 가장 늦게 방문하게 되는 정점은?(단, 조건이 같으면 알파벳 순서가 빠른 것을 우선 방문한다.)(2004 중등)

① C
② D
③ E
④ F
⑤ G



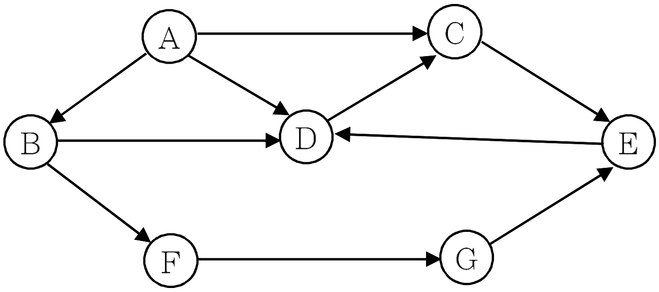
<풀이> ③

너비우선탐색의 결과는 A, B, D, C, F, G, E이다.

문제Ⅲ.4.15

다음과 같은 방향그래프의 정점 A에서 너비우선탐색(BFS, breadth-first search)을 할 때 가장 늦게 방문하게 되는 정점은?(단, 조건이 같으면 알파벳 순서가 빠른 것을 우선 방문한다.)(2004 중등)

① C
② D
③ E
④ F
⑤ G



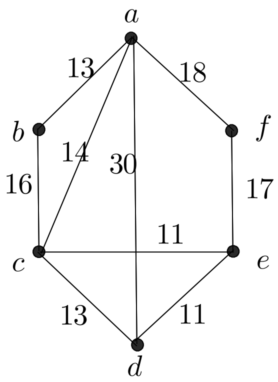
<풀이> ⑤
너비우선탐색의 결과는 A, B, C, D, F, E, G이다.

라. 최단경로 알고리즘

가중그래프에서 두 정점사이에 최단 경로를 구하는 여러 가지 알고리즘 중에 모든 가중치가 양수일 경우에 적용하기가 쉬운 다익스트라(Dijkstra) 알고리즘이 있다. 다익스트라 알고리즘을 구현하기 위한 가중그래프의 인접행렬 $W[x,y]$는 간선의 가중치로 행렬의 원소를 결정한다. 즉,

$$W[x,y]=\begin{cases} 0, & x=y \\ \infty, & x\text{와 } y\text{ 를 연결하는 간선이 없으면} \\ w, & \text{그외의 경우} \end{cases}$$

이다. ∞는 그래프에서 어떠한 가중치보다도 큰 값을 의미한다. 〔그림 Ⅲ.4.15〕는 가중그래프를 인접행렬로 표현하였다.



	a	b	c	d	e	f
a	0	13	14	30	∞	18
b	13	0	16	∞	∞	∞
c	14	16	0	13	11	∞
d	30	∞	13	0	11	∞
e	∞	∞	11	11	0	17
f	18	∞	∞	∞	17	0

[그림Ⅲ.4.15] 가중그래프와 인접 행렬

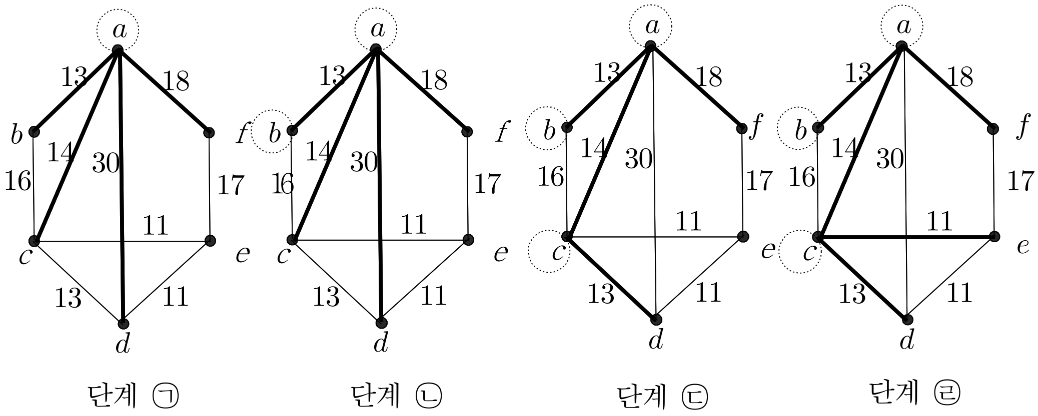
다익스트라의 알고리즘으로 정점 a에서 나머지 정점들까지 최단경로를 구하는 과정은 다음과 같다.

① a를 정점 집합 S에 넣고 정점 a에서 각 정점까지의 거리를 계산한다.
② S에 속하지 않은 정점들 중에서 최소 가중치를 갖는 정점을 S에 넣고 a에서 나머지 정점들에 대해 바로 가는 비용보다 S에 속한 정점들을 경유해서 가는 것이 비용이 더 작게 드는 경로를 찾아서 그 정점까지의 더 작은 비용의 경로로 바꾼다.
③ a에서 모든 정점 A에 추가될 때까지 과정②를 반복한다.

이제 a를 출발점으로 다른 정점으로 가는 최단경로를 구하기 위해 출발점 a에서 다른 정점으로 가는 최소 비용을 배열에 저장한다. 맨 처음 집합 S에는 a만 들어있다. 그리고 배열은 정점 a에서 다른 모든 정점들까지 직접 연결된 거리를 가진다. S에 속하지 않은 정점들 중에서 최소 가중치를 갖는 정점을 계산한다. 여기서는 정점 b이며 a에서 나머지 정점들에 대해 바로 가는 정점보다 b를 경유해서 가는 것이 더 비용이 작게 드는 경로를 찾는다. b를 S에 넣고 더 작게 되는 경로가 없으므로 다음으로 S에 속하지 않은 정점들 중에서 최소 가중치를 갖는 정점을 계산한다. 정점 c를 찾을 수 있고 c를 S에 넣고 a에서 나머지 정점들에 대해 바로 가는 정점보다 c를 경유해서 가는 것이 더 비용이 작게 드는 경로를 찾는다. 여기서 a에서 c를 경유하는 두 개의 노드 d와 e까지의 더 짧은 경로를 찾을 수 있다. 단계 ㉣에서 a에서 모든 다른 정점들까지의 최단 경로를 구할 수 있었다. 〈표Ⅲ.4.1〉과 〔그림Ⅲ.4.16〕은 초기화인 단계 ㉠부터 최종 단계 ㉣까지를 나타낸 것이다.

〈표Ⅲ.4.1〉 노드 a에서의 거리

단계 \ 정점	a	b	c	d	e	f
㉠	0	13	14	30	∞	18
㉡	0	13	14	30	∞	18
㉢	0	13	14	27	∞	18
㉣	0	13	14	27	25	18

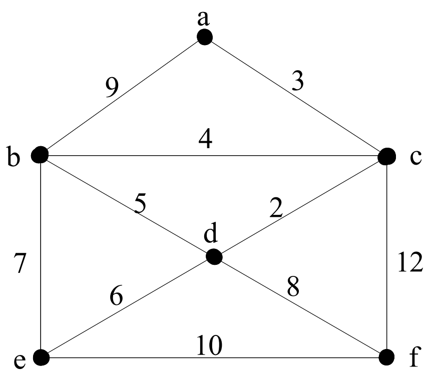


[그림Ⅲ.4.16] 정점 a에서 각 정점까지의 최단거리

예제Ⅲ.4.16

다음 가중그래프에서 a에서 각 정점까지 가는데 지불해야 하는 최소비용이 잘못 연결된 것은?

① b, 7
② c, 3
③ d, 5
④ e, 11
⑤ f, 15

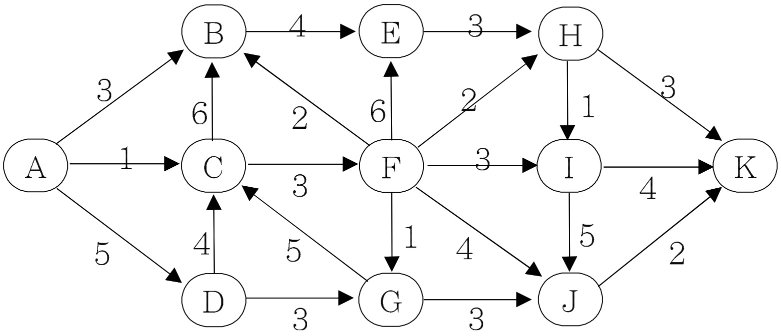


<풀이> ⑤

단계 \ 정점	a	b	c	d	e	f
㉠	0	9	3	∞	∞	∞
㉡	0	7	3	5	∞	15
㉢	0	7	3	5	11	13

문제Ⅲ.4.16

다음은 각 지점을 연결하는 도로 상황을 나타내는 그림이다. 각 도로는 화살표를 따라 일방통행만 가능하며 화살표 위에는 도로 이용할 때 지불해야 하는 비용이 쓰여 있다. A지점부터 K지점까지 가는데 소요되는 최소 비용은 얼마인가?(2007 초등)



① 8　　② 9　　③ 10　　④ 11　　⑤ 12

<풀이> ②

단계 \ 정점	A	B	C	D	E	F	G	H	I	J	K
㉠	0	3	1	5	∞	∞	∞	∞	∞	∞	∞
㉡	0	3	1	5	∞	4	∞	∞	∞	∞	∞
㉢	0	3	1	5	7	4	∞	∞	∞	∞	∞
㉣	0	3	1	5	7	4	5	6	7	8	∞
㉤	0	3	1	5	7	4	5	6	7	8	9

연 습 문 제

1 10개의 정점을 갖는 그래프에서 각 정점의 차수가 6이라면 몇 개의 간선이 존재하는가?

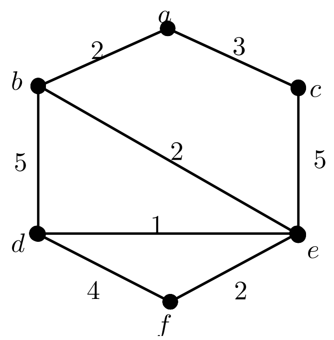
① 10개 ② 20개 ③ 30개 ④ 60개 ⑤ 120개

2 루프가 없고 두 정점을 연결하는 간선이 하나뿐인 그래프를 단순그래프(simple graph)라고 한다. 다섯 개의 정점에 대한 다음 각 정점의 차수가 주어졌을 때, 단순 그래프가 아닌 것은?

① 2,2,2,2,2 ② 1,2,3,3,3 ③ 2,2,2,3,3
④ 3,3,3,3,4 ⑤ 1,1,1,1,4

3 모든 정점의 차수가 같은 그래프를 정규 그래프라고 한다. 15개의 간선을 갖고 차수가 5인 정규그래프의 정점의 수를 구하시오.

① 5개 ② 6개 ③ 8개 ④ 10개 ⑤ 15개

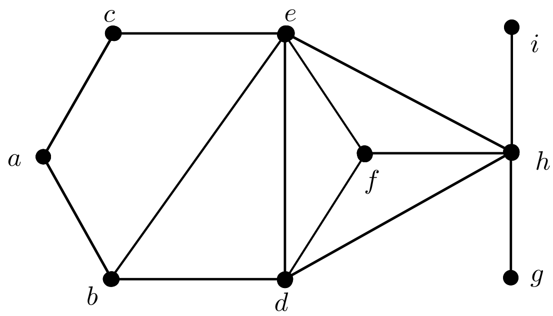


4 다음 가중그래프에서 a와 f사이의 최단경로는 얼마인가?

① 6 ② 8 ③ 9
④ 10 ⑤ 11

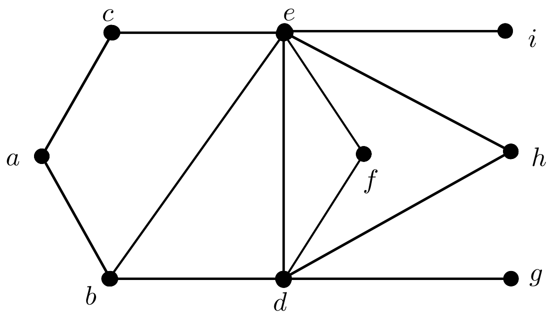
5. 다음 그래프의 정점 a에서 깊이우선탐색 방법으로 탐색한 결과에서 가장 마지막으로 방문하게 되는 정점은?(단, 조건이 같으면 알파벳 순서가 빠른 것을 우선 방문한다.)

① e
② f
③ h
④ i
⑤ g



6. 다음 그래프의 정점 a에서 너비우선탐색 방법으로 탐색한 결과에서 가장 마지막으로 방문하게 되는 정점은?(단, 조건이 같으면 알파벳 순서가 빠른 것을 우선 방문한다.)

① e
② f
③ h
④ i
⑤ g



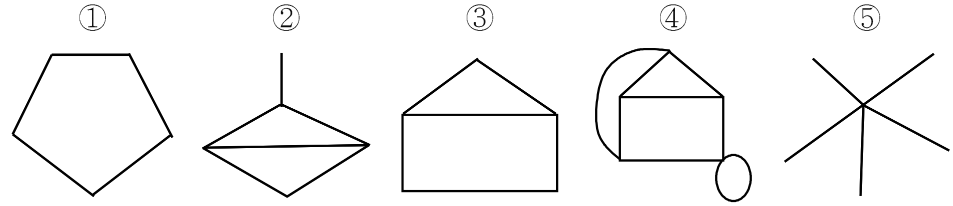
풀 이

1 ③

정점의 차수의 합은 간선의 수의 2배이므로, 간선의 수는 30개이다.

2 ④

주어진 간선의 개수로 그린 그래프는 다음과 같다. ④는 루프를 가져야 하므로 단순그래프가 아니다.



3 ②

간선의 개수는 $\frac{1}{2}nr$이므로 $\frac{1}{2}\times n\times 5=15$이므로 $n=6$이다.

4 ①

〈표Ⅲ.4.3〉 노드 a에서의 거리

단계 \ 정점	a	b	c	d	e	f
㉠ a	0	2	3	∞	∞	∞
㉡ a, b	0	2	3	5	4	∞
㉢ a, b, e	0	2	3	5	4	6

5 ④

깊이우선탐색의 결과는 a, b, d, e, c, f, h, g, i이다.

6 ④

너비우선탐색의 결과는 a, b, c, d, e, f, g, h, i이다.

5. 트리

순환이 없는 연결그래프를 트리(tree)라고 한다. 트리는 정보가 간선으로 연결되는 비선형 자료 구조이다. 트리 이론은 컴퓨터 공학에서 효율적인 알고리즘의 구축과 분산된 컴퓨터 간의 연결, 자료 처리, 자료의 정렬과 전달을 위한 효율적인 프로그램 등을 구성하는데 유용하게 이용된다.

가. 트리

트리는 루트(root)라고 불리는 노드(node)를 갖는 순환되지 않는 연결그래프이다. 그래프의 정점을 트리에서는 노드라고 한다. 트리는 어느 간선을 제거해도 비연결그래프가 되고, 서로 다른 두 꼭지점 사이에 유일한 단순 경로가 존재하는 그래프이다. 트리의 루트는 시작 노드로 트리의 최상위에 위치한다. n개의 노드를 갖는 트리는 $n-1$개의 간선으로 구성되어 있으며, 간선을 제거하거나 추가하면 트리의 성질을 잃게 된다.

예제Ⅲ.5.1

트리 모양의 연결된(connected) 그래프가 n개의 정점(vertex)을 가질 때, 간선(edge)의 수는?(2003 중등)

① $n-1$개 ② n개 ③ $n+1$개 ④ $2n-1$개 ⑤ $2n$개

<풀이> ①

트리에서 간선의 수는 정점의 수보다 1이 작다. 즉, 정점의 수가 n개이면 간선의 수는 $n-1$이다.

문제Ⅲ.5.1

트리 모양의 연결된(connected) 그래프가 10개의 정점(vertex)을 가질 때, 간선(edge)의 수는?(2003 초등)

① 8 ② 9 ③ 10 ④ 11 ⑤ 12

<풀이> ②

트리에서 간선의 수는 정점의 수보다 1이 작으므로 간선의 수는 9개가 된다.

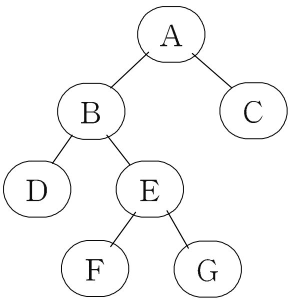
그래프 G의 부분그래프 G_1이 트리이고 G_1이 G의 모든 노드를 포함하고 있다면 G_1을 G의 신장트리(spanning tree)라고 한다. 또한 그래프 G의 간선에 임의의 값이 할당된 가중그래프(weighted graph)가 있을 때, 여러 개의 신장트리를 만들 수 있는데 이러한 신장트리들 중에서 가중치의 합이 가장 작은 신장트리를 최소비용신장트리(minimal cost spanning tree)라고 한다.

노드의 차수(degree)는 그 노드의 서브트리의 개수이며 트리의 차수란 그 트리에 있는 노드의 최대 차수이다. 임의의 노드의 서브트리를 그 노드의 자식노드(children node)라고 하고 서브트리를 갖는 노드를 그 서브트리의 부모노드(parent node)라고 한다. 동일한 부모노드를 가진 노드를 형제노드(sibling node)라고 한다. 트리에서 자식노드를 갖지 않은 노드를 단말노드(leaf node)라고 하고 단말노드가 아닌 노드를 내부노드(internal node)라고 한다. 루트로부터 그 노드에 이르는 경로에 나타난 모든 노드를 조상노드(ancestors node)라고 하며 자손노드(descendant node)는 그 노드로부터 단말노드에 이르는 경로 상에 나타난 모든 노드를 말한다. 서로 교차하지 않는 트리의 집합으로 트리에서 루트를 제거하면 숲(forest)이 된다.

예제Ⅲ.5.2

다음 트리에서 단말노드(terminal node)의 개수는?

① 2
② 3
③ 4
④ 5
⑤ 6



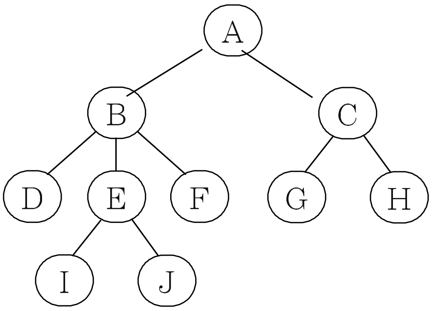
<풀이> ③

단말노드란 자식노드를 갖지 않는 노드이므로 D, F, G, C이다.

문제Ⅲ.5.2

다음 트리에서 노드 E의 형제노드(sibling node)는 몇 개인가?(2004 중등)

① 1개
② 2개
③ 3개
④ 4개
⑤ 5개



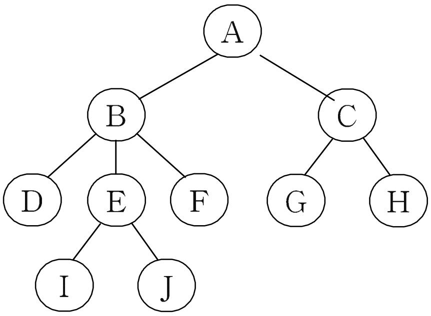
<풀이> ②

E의 형제노드는 E의 부모노드 B의 자식노드인 D, F이므로 2개이다.

예제Ⅲ.5.3

다음 트리에서 노드 B의 자식노드(child node)는 몇 개인가?(2004 초등)

① 1개
② 2개
③ 3개
④ 4개
⑤ 5개



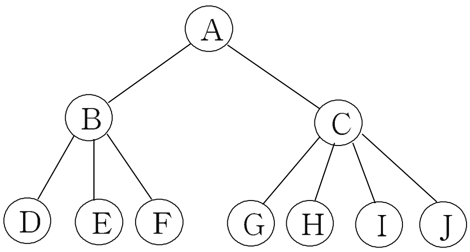
<풀이> ③

B의 자식노드는 D, E, F이므로 3개이다.

문제Ⅲ.5.3

아래 그림과 같은 트리의 차수는? (2006 중고등)

① 2
② 3
③ 4
④ 7
⑤ 10



<풀이> ③

C의 서브노드 개수가 4개로 가장 많으므로 이 트리의 차수는 4이다.

이 책에서는 루트의 레벨을 1로 가정한다.(책에 따라 루트의 레벨을 0이라고 가정하기도 한다.) 각 노드의 레벨(level)은 자손노드로 내려가면서 그 부모노드의 레벨에 1을 더한다. 트리의 높이(깊이, height, depth)는 노드의 레벨중에서 최대값이다.

예제Ⅲ.5.4

트리(tree)에서는 어떤 노드(node)에서 루트(root)까지 가는 데 자신을 포함하여 거쳐 가는 노드의 수를 레벨(level)이라고 한다. 그렇다면 다음 트리에서 아래 보기 중 레벨이 가장 큰 노드는? (2005 초등)

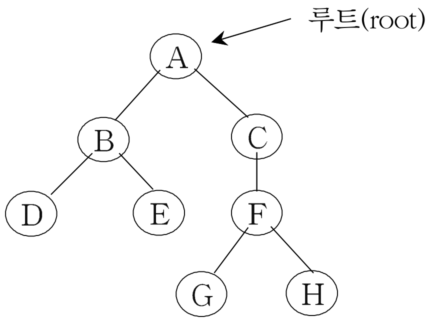
① A

② C

③ D

④ F

⑤ G



<풀이> ⑤

루트 노드의 레벨을 1이라고 가정하면 노드 G, H의 레벨이 4로 가장 크다.

문제Ⅲ.5.4

다음 중 트리에 대하여 설명이 **가장 적당하지 <u>않은</u> 것은?** (2006 교원)

① 모든 노드는 루트 노드에서 출발한다.

② 트리들을 합하면 포리스트(숲, forest)가 된다.

③ 계층적인 구조를 갖고 있다.

④ 선형구조를 나타내기 알맞다.

⑤ 비순환, 연결그래프다.

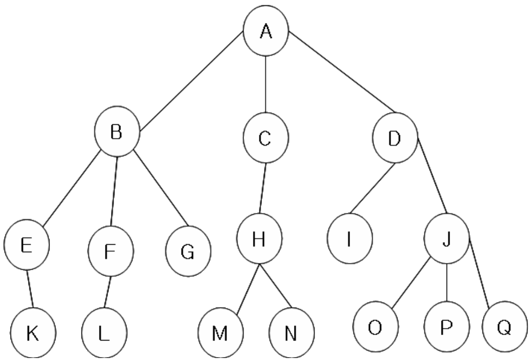
<풀이> ④

트리는 비선형구조이므로 선형구조를 나타내기에는 적당하지 않다.

예제Ⅲ.5.5

다음 트리에 대한 설명이 **<u>틀린 것은</u>** ? (2007 교원)

① 노드 H의 부모 노드는 C 이다
② 노드 D의 형제노드는 B, C 이다
③ 이 트리의 차수는 4 이다.
④ 이 트리의 깊이는 4 이다.
⑤ 노드 N의 조상은 H, C, A 이다.



<풀이> ③

이 트리의 차수는 3이고 이 트리의 깊이는 4이다.

문제Ⅲ.5.5

예제Ⅲ.5.5의 트리에서 단말노드의 개수는?

① 7개 ② 8개 ③ 9개 ④ 10개 ⑤ 11개

〈풀이〉 ③

단말노드는 K, L, G, M, N, I, O, P, Q이다.

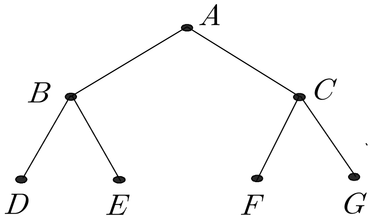
나. 이진트리

1) 이진트리

트리 중에서 가장 많이 활용되는 트리는 이진트리(binary tree)인데 각 노드는 반드시 2개 이하의 자식노드를 가져야 한다. 이진트리는 노드의 수가 0이 될 수 있고, 각 노드의 자식노드는 왼쪽 자식노드와 오른쪽 자식노드로 구분되어지며 순서가 있다.

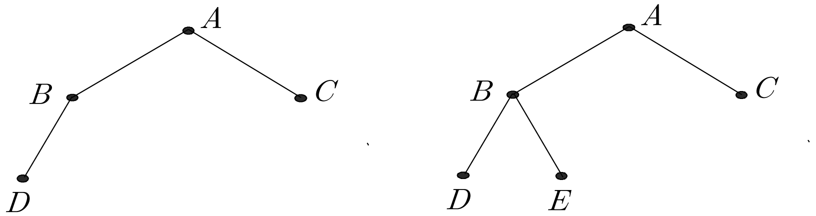
경사이진트리(skewed binary tree)에는 각 노드가 부모의 왼쪽 자식만 가지는 왼쪽경사이진트리와 각 노드가 부모의 오른쪽 자식만 가지는 오른쪽경사이진트리가 있다.

포화이진트리(full binary tree)는 모든 내부 노드가 2개의 자식노드를 가지고, 모든 단말노드는 같은 깊이를 갖는 이진트리이다. 따라서 포화이진트리는 깊이가 k이면, 전체 노드수가 2^k-1인 이진트리이다. 〔그림Ⅲ.5.1〕은 깊이가 3이고 전체 노드수가 $2^3-1=7$인 포화이진트리의 예를 보여준다.



[그림Ⅲ.5.1] 포화이진트리

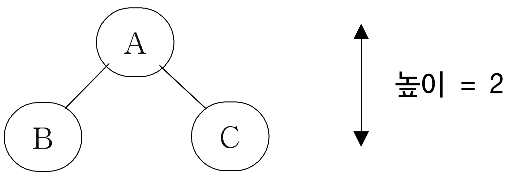
완전이진트리(complete binary tree)는 각 레벨에서 왼쪽부터 오른쪽으로 노드들이 차례대로 채워진 이진트리이다. 〔그림Ⅲ.5.2〕는 깊이가 3인 완전이진트리의 예를 보여준다.



[그림Ⅲ.5.2] 완전이진트리

예제Ⅲ.5.6

다음 트리는 높이(height)가 2인 트리이다. 노드(node)의 개수가 1000개인 완전이진트리(complete binary tree)의 높이는 얼마인가?(2003 중등)



① 9 ② 10 ③ 11 ④ 12 ⑤ 13

<풀이> ②

노드의 개수가 1000개인 포화이진트리의 높이는 다음과 같이 구할 수 있다.

$512 < 2^k - 1 \leqq 1024$

$513 < 2^k \leqq 1025$

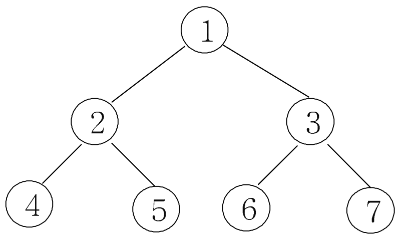
$\log_2 513 < \log_2 2^k \leqq \log_2 1025$

$\therefore 9 < k \leqq 10$

문제Ⅲ.5.6

높이가 3인 포화이진트리는 아래와 같이 7개의 노드(node)로 이루어져 있다. 그렇다면 높이가 7인 포화이진트리는 몇 개의 노드로 이루어져 있을까? (2005 초등)

① 63개
② 64개
③ 127개
④ 128개
⑤ 255개



<풀이> ③

$2^7 - 1 = 128 - 1 = 127$ 이다.

예제Ⅲ.5.7

깊이가 k인 이진트리가 가질 수 있는 최대 노드 수를 구하시오.

<풀이> $2^k - 1$

깊이가 k인 이진트리가 가질 수 있는 최대 노드 수는 $\sum_{i=1}^{k} 2^{i-1}$ 이므로 $2^k - 1$ 이다.

문제Ⅲ.5.7

자식노드가 2개 이하인 트리를 이진트리라고 한다. 루트 노드의 레벨이 1이라고 할 때 최대 레벨이 10인 이진트리가 가질 수 있는 최대 노드 수는 얼마인가?(2005 교원)

① 511 ② 512 ③ 1023 ④ 1024 ⑤ 2048

<풀이> ③

최대 레벨이 10인 이진트리가 가질 수 있는 최대 노드 수는 $2^{10}-1=1024-1=1023$개 이다.

예제Ⅲ.5.8

이진트리에서 단말노드수를 n_0, 차수가 2인 노드수를 n_2라고 하면 $n_0=n_2+1$가 성립한다. 이를 증명하시오.

<풀이>

n_1을 차수가 1인 노드라고 하고 n을 이진트리의 전체 노드 수라고 하면 $n=n_0+n_1+n_2$이다. 트리에서 간선의 수 b는 노드 수 n보다 1이 작으므로 $b=n-1$이 성립한다. 차수가 2인 노드에서는 2개의 간선이 나오고, 차수가 1인 노드에서는 1개의 간선이 생기므로 $b=2n_2+n_1$이다. 따라서 $n-1=2n_2+n_1$이 되고 $n_0+n_1+n_2=1+n_1+2n_2$이므로 $n_0=n_2+1$이다.

문제Ⅲ.5.8

이진트리에서 차수가 1인 노드 수가 10개이고 차수가 2인 노드 수가 12개이면 전체 노드 수는 모두 몇 개인가?

① 12개 ② 13개 ③ 30개 ④ 35개 ⑤ 36개

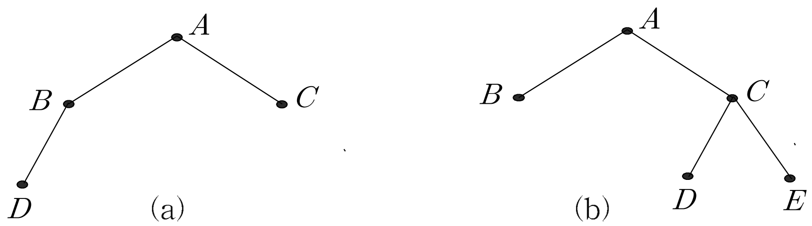
<풀이> ④

$n_0=n_2+1$이므로 $n_2=12$이고 $n_0=13$이므로 전체 노드 수 $n=n_0+n_1+n_2=13+10+12=35$이다.

2) 이진트리의 표현 방법

이진트리의 배열을 이용한 표현 방법은 다음과 같다. 먼저 일차원 배열을

생각하면 트리의 각 노드에 루트 노드를 1로 시작해서 왼쪽에서 오른쪽 방향으로 번호를 붙인다. 이때 각 노드의 숫자와 배열의 인덱스를 일치시키는 방법은 사용하면 〔그림Ⅲ.5.3〕의 왼쪽 트리를 〈표Ⅲ.5.4〉의 일차원 배열로 표현할 수 있다.



[그림Ⅲ.5.3] 이진트리

(a)

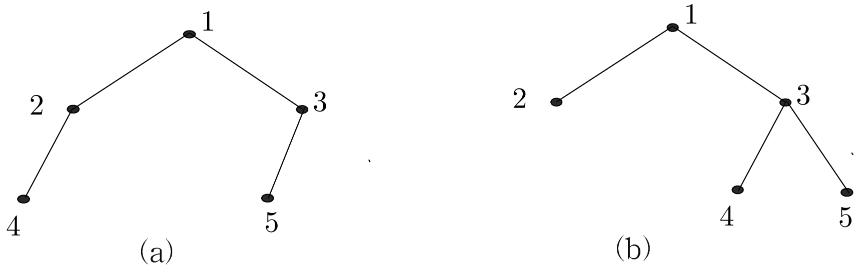
[1]	[2]	[3]	[4]	[5]	[6]	[7]
A	B	C	D			

(b)

[1]	[2]	[3]	[4]	[5]	[6]	[7]
A	B	C			D	E

[그림Ⅲ.5.4] 이진트리의 일차원 배열 표현

이진트리를 이차원 배열로 표현하면 다음과 같다. 행은 여러 개의 노드 값을 가지고 열은 왼쪽 자식노드 값과 오른쪽 자식노드 값을 가지는 이차원 배열을 만든다. 자식노드가 없는 경우는 그 값을 0으로 한다.
〔그림Ⅲ.5.6〕은 〔그림Ⅲ.5.5〕의 이진트리의 이차원 배열 표현이다.



[그림Ⅲ.5.5] 이진트리

	왼쪽 자식	오른쪽 자식
1	2	3
2	4	0
3	5	0
4	0	0
5	0	0

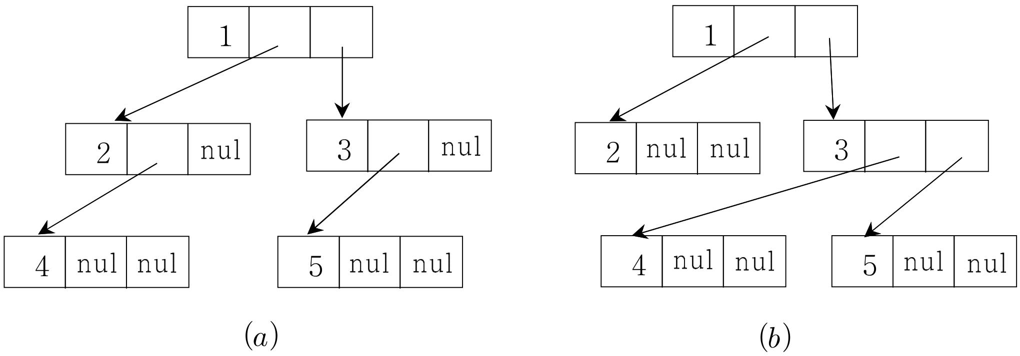
(a)

	왼쪽 자식	오른쪽 자식
1	2	3
2	0	0
3	4	5
4	0	0
5	0	0

(b)

[그림Ⅲ.5.6] 이진트리의 이차원 배열 표현

이진트리의 연결리스트를 이용한 표현 방법은 다음과 같다. 연결리스트의 각 노드는 노드값, 왼쪽 포인터, 오른쪽 포인터인 세 개의 필드로 되어 있다. 왼쪽 포인터는 왼쪽 자식노드를, 오른쪽 포인터는 오른쪽 자식노드를 가리킨다. 자식노드가 없는 단말노드의 포인터들은 항상 NULL값을 갖는다. 〔그림Ⅲ.5.7〕은 〔그림Ⅲ.5.5〕의 이진트리를 연결리스트로 표현한 것이다.

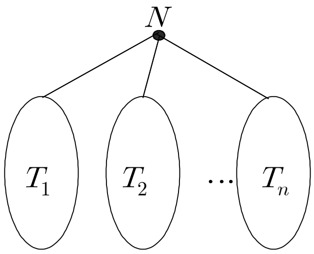


[그림Ⅲ.5.7] 이진트리의 연결리스트 표현

다. 트리 순회

1) 이진트리 순회

트리 순회(tree traverse)란 어떤 정해진 순서에 의해 각 노드들을 한 번씩만 방문함으로써 모든 노드를 순회하는 것을 의미한다. 트리에서 사용되는 순회 알고리즘에는 전위순회(preorder traverse), 중위순회(inorder traverse), 후위순회(postorder traverse)가 있다.

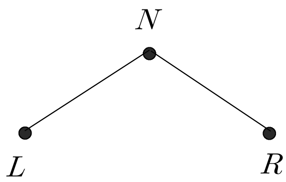


[그림Ⅲ.5.8] 트리

[그림Ⅲ.5.8]처럼 트리의 루트 노드가 N이고 $T_1, T_2, \cdots, T_n$ 을 서브트리로 가진다면, 전위순회는 루트 노드 N을 우선 방문하고 각 서브트리 $T_1, T_2, \cdots, T_n$ 들을 가장 왼쪽 서브트리에서 가장 오른쪽 서브트리로 차례대로 이동하면서 방문해 나가는 방법이다. 중위순회는 루트의 가장 왼쪽 서브트리 T_1을 우선 방문하고, 루트 노드 N을 방문한 다음, 나머지 각 서브트리 $T_2, \cdots, T_n$ 들을 오른쪽 서브트리로 이동하면서 방문한다. 마지막으로 후위순회는 루트의 각 서브트리 $T_1, T_2, \cdots, T_n$ 들을 가장 왼쪽 서브트리에서 가장 오른쪽 서브트리로 차례로 이동하면서 방문한 후에, 루트 노드 N을 맨 마지막에 방문한다.

순회는 서브트리를 만나게 되면 각 서브트리에 대해 해당 순회과정을 순환적으로 적용한다. 트리 순회는 이진트리인 경우에 더 쉽게 적용될 수 있다.

루트가 N이고 왼쪽 서브트리 L가, 오른쪽 서브트리가 R인 이진트리가 〔그림Ⅲ.5.9〕에 표시되어 있다.



[그림Ⅲ.5.9] 이진트리

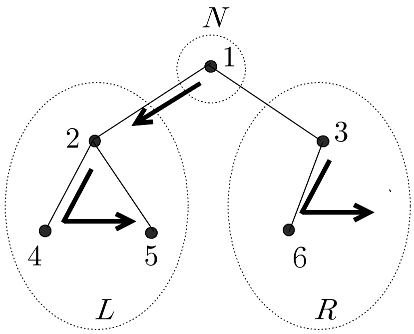
가능한 경우는 $N-L-R$, $N-R-L$, $L-N-R$, $L-R-N$, $R-N-L$, $R-L-N$이 있는데, 왼쪽을 오른쪽보다 우선 방문한다고 가정하면 $N-L-R$, $L-N-R$, $L-R-N$ 세 가지 경우가 된다.

가) 전위순회(루트→왼쪽→오른쪽)

① 루트를 방문한다.

② 왼쪽 서브트리를 전위순회한다.

③ 오른쪽 서브트리를 전위순회한다.



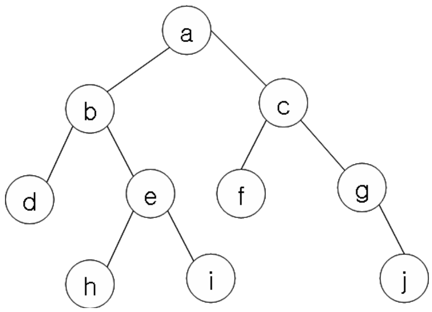
[그림Ⅲ.5.10] 전위순회

[그림Ⅲ.5.10]을 보면 노드 1의 서브트리는 2개가 있다. 루트 노드인 1을 우선 방문하고 왼쪽 서브트리의 루트 노드 2를 방문한 후 왼쪽 자식노드 4, 오른쪽 자식노드 5를 방문한다. 오른쪽 서브트리로 가서 노드 3을 방문하고 마지막으로 왼쪽 노드 6을 방문한다. 그래서 전위순회의 결과는 1, 2, 4, 5, 3, 6이다.

예제Ⅲ.5.9

전위표기법은 트리의 루트를 우선 방문한 후에 각 서브트리들을 전위순회를 이용하여 가장 왼쪽 서브트리부터 가장 오른쪽 서브트리로 차례대로 이동하면서 방문하는 방법이다. 다음 이진트리에서 전위표기법을 올바르게 구한 것은?

① a → b → d → e → h → i → c → f → g → j
② d → b → h → e → i → a → f → c → g → j
③ d → h → i → e → b → f → j → g → c → a
④ a → b → c → d → e → f → g → h → i → j
⑤ a → d → h → i → e → f → g → j → g → c

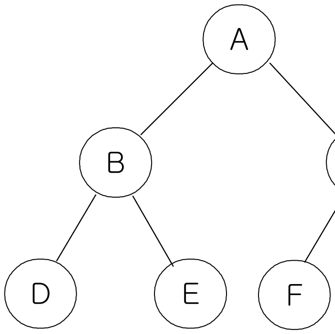


<풀이> ①

전위순회의 결과는 a → b → d → e → h → i → c → f → g → j이다.

문제Ⅲ.5.9

다음 트리를 전위 순회(preorder traversal)했을 때 방문하는 노드를 순서대로 나열한 것은?(2007 중고등)



① A B C D E F　　② A B D E C F　　③ D B E A F C
④ D E B F C A　　⑤ D E F B C A

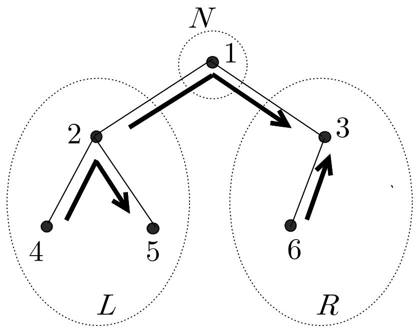
<풀이> ②

전위순회의 결과 A, B, D, E, C, F이다.
중위순회의 결과 D, B, E, A, F, C이다.
후위순회의 결과 D, E, B, F, C, A이다.

나) 중위순회(왼쪽→루트→오른쪽)

① 왼쪽 서브트리를 중위순회한다.
② 루트를 방문한다.
③ 오른쪽 서브트리를 중위순회한다.

중위순회는 왼쪽 서브트리를 우선 방문하고 루트 노드를 방문한 후 오른쪽 서브트리를 방문한다.



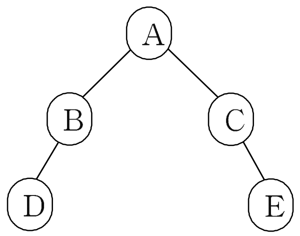
[그림Ⅲ.5.11] 중위순회

[그림Ⅲ.5.11]에서 왼쪽 서브트리의 가장 왼쪽 자식노드인 노드 4를 가장 먼저 방문하고 노드 2를 방문하고 오른쪽 자식인 노드 5를 방문한다. 루트 노드 1을 방문하고 오른쪽 서브트리의 왼쪽 노드 6을 방문한 후 마지막으로 노드 3을 방문한다. 그래서 중위순회의 결과는 4, 2, 5, 1, 6, 3순이다.

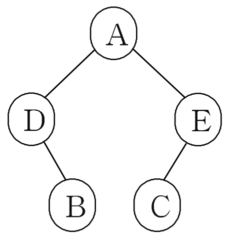
예제Ⅲ.5.10

다음 중 트리를 중위순회(inorder traversal)한 결과가 다른 하나는? (2004고등)

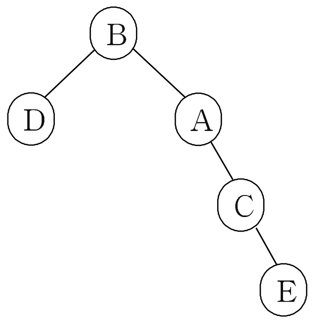
①



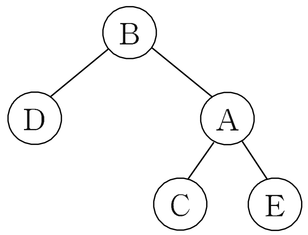
②



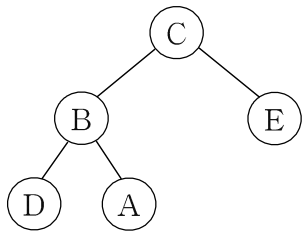
③



④



⑤



<풀이> ④

①, ②, ③과 ⑤의 중위순회의 결과는 D, B, A, C, E이고 ④의 중위순회의 결과는 D, B, C, A, E이다.

문제Ⅲ.5.10

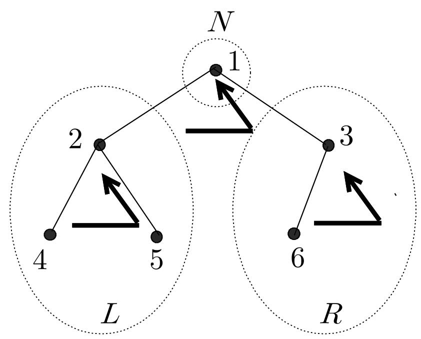
다) 후위순회(왼쪽→오른쪽→루트)

① 왼쪽 서브트리를 후위순회한다.

② 오른쪽 서브트리를 후위순회한다.

③ 루트를 방문한다.

[그림Ⅲ.5.12]에서 왼쪽 서브트리의 가장 왼쪽 노드인 노드 4를 가장 먼저 방문하고 오른쪽 자식노드 5, 노드 2를 방문하게 된다.



[그림Ⅲ.5.12] 후위순회

그 다음 오른쪽 서브트리로 가서 왼쪽 자식노드 6을 방문하고 오른쪽 자식노드가 없으므로 노드 3을 방문하고 마지막으로 노드 1을 방문한다. 그래서 후위순회의 결과는 4, 5, 2, 6, 3, 1순이다.

예제Ⅲ.5.11

다음과 같은 이진트리를 후위순회(Postorder Traversal)할 때 지나는 노드의 순서로 옳은 것은? (2002 초등)

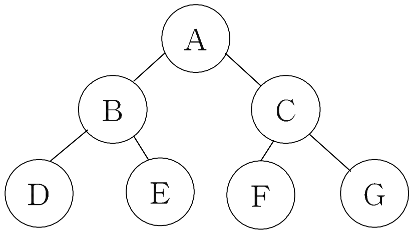
① A-B-C-D-E-F-G

② A-C-B-G-F-E-D

③ D-B-E-A-F-C-G

④ D-E-B-F-G-C-A

⑤ G-F-C-E-D-B-A



<풀이> ④

전위 순위의 결과는 A, B, D, E, C, F, G이다.
중위순회의 결과는 D, B, E, A, F, C, G이다.
후위순회의 결과는 D, E, B, F, G, C, A이다.

문제Ⅲ.5.11

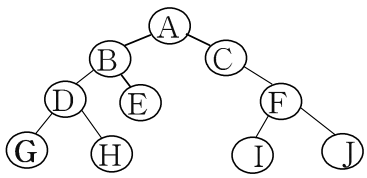
어떤 이진트리를 전위순회(preorder traversal)한 결과가 ABDGHECFIJ이고, 중위순회(inorder traversal)한 결과는 GDHBEACIFJ이었다. 이 이진트리를 후위순회(postorder traversal)한 결과가 올바른 것은?
(2004 교원)

① ABCDEFGHIJ ② GHDEBIJFCA ③ HDGBEJICFA

④ CFIJBDGHEA ⑤ EBCDGHJIFA

<풀이> ②

후위순회의 결과는 G, H, D, E, B, I, J, F, C, A 이다. 루트노드가 A임을 알 수 있고 A를 기준으로 GDHBE는 A의 왼쪽서브트리가 된다.



예제Ⅲ.5.12

A부터 G가 적혀있는 노드로 이루어진 이진트리가 있다. 이 이진트리에서 전위순회로 방문한 노드의 순서가 ABDECFGH, 중위 순위로 방문한 노드의 순서가 DBEAGFHC일 때 후위순회로 방문한 노드의 순서는? (2005 중고등)

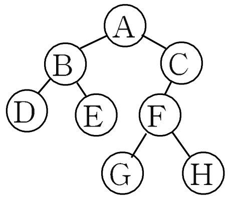
① ACFGHBDE ② ACFHGBED

③ DBEGHFCA ④ DEBGHFCA

⑤ HGFCEDBA

<풀이> ④

전위순회와 중위순회를 이용하여 트리를 그려보면



문제Ⅲ.5.12

후순위(후위, Postorder) 순회의 원리는 특정 노드에서 자신의 왼쪽 서브트리와 오른쪽 서브트리를 차례로 방문한 후, 마지막으로 자신의 노드를 방문한다. 이 원리를 모든 노드에 재귀적으로 적용하면 각 노드를 한번씩 방문할 수 있다. 아래의 이진트리에서 노드의 방문순서를 바르게 나열한 것은?(2006 교원)

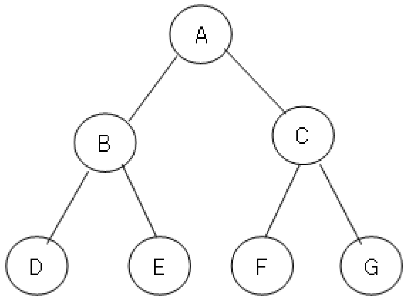
① A B C D E F G

② D E B F G C A

③ A B D E C F G

④ B D A E F C G

⑤ G F E D C B A



<풀이> ②

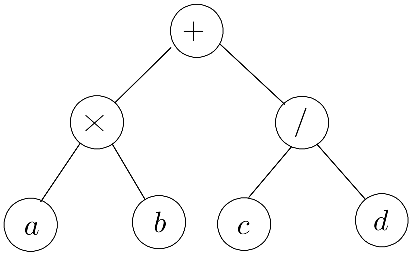
전위순위의 결과는 A, B, D, E, C, F, G이다.

중위순회의 결과는 D, B, E, A, F, C, G이다.

후위순회의 결과는 D, E, B, F, G, C, A이다.

라. 수식트리

수식을 트리로 표현한 트리를 수식트리라고 하는데 연산자의 위치에 따라 중위(infix), 후위(postfix), 전위(prefix)표기가 있다. 우리는 중위표기 수식을 쓴다. 예를 들면, $a \times b + c/d$이다. 이것은 연산자 우선 순위를 고려하면 $(a \times b) + (c/d)$이므로 [그림Ⅲ.5.13]에 수식트리로 표현할 수 있다. 그래프나 트리를 그리는 경우, 간혹 노드를 원으로 표시하고 원 안에 노드 값을 두기도 한다.



[그림Ⅲ.5.13] 수식트리

중위표기는 수식을 표현할 때 보편적으로 사용하는 방식이기는 하지만 연산자 우선 순위를 고려하여 괄호를 제대로 사용하여야 원하는 연산의 결과를 얻을 수 있다. 그러나 전위표기와 후위표기를 사용하면 괄호를 사용하지 않고도 수식을 처리할 수 있다는 장점이 있다.

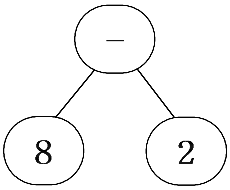
〔그림Ⅲ.5.13〕을 후위표기법으로 읽으면 $a\,b \times c\,d\,/ +$이 된다. 후위표기에서는 연산자가 피연산자의 뒤에 위치하게 된다.

전위표기법으로 읽으면 $+ \times a\,b\,/\,c\,d$가 된다. 전위표기에서는 연산자가 피연산자의 앞에 위치하게 된다. 컴퓨터에서 수식을 처리할 경우는 괄호가 없는 전위나 후위표기가 훨씬 수월하다.

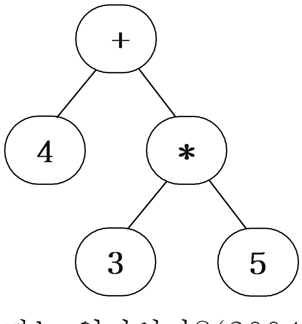
예제Ⅲ.5.13

사칙연산으로 이루어진 수식은 우선순위에 따라 다음 예와 같이 이진트리로 표현할 수 있다.

〈 8-2의 예 〉



〈 4+3*5의 예 〉



그렇다면 다음 트리로 표현된 수식을 계산한 결과는 얼마인가?(2004 초등)

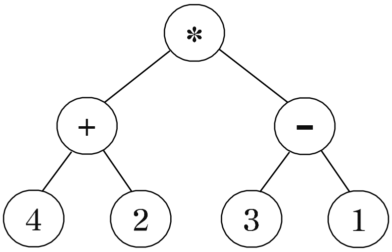
① 8

② 9

③ 12

④ 18

⑤ 23



<풀이> ③

(4+2)*(3-1)=6*2=12

문제Ⅲ.5.13

후위 표기식은 연산자가 두 개의 피연산자 사이에 나오는 중위 표기식과 달리 연산자가 두 개의 피연산자 뒤에 나온다. 예를 들어 왼쪽의 중위 표기식은 오른쪽의 후위 표기식과 같은 식이다.(2002 초등)

중위 표기식	후위 표기식
1+2	12+
2×3+4	23×4+

그렇다면 다음 후위 표기식을 계산한 결과는 얼마인가?

235×+

① 11　② 13　③ 16　④ 17　⑤ 25

<풀이> ④

피연산자 두 개와 연산자를 묶어서 바로 계산하는 것이 가장 빠르다.

2 3 5 × + = 2 (3 5 ×) + = 2 15 + = 17

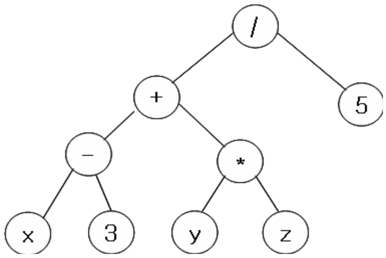
두 번째로 수식트리를 만들어 중위표기를 이용하여 계산한다.

예제Ⅲ.5.14

이진트리를 순회하는 표준적인 방법에는 전위(preorder), 중위(inorder), 후위(postorder)의 3가지 방법이 있다. 이는 루트와 왼쪽 서브트리, 오른쪽 서브트리 중에서 루트를 언제 방문하느냐에 따라 구분된다. 즉 루트를 서브트리에 앞서서 먼저 방문하면 전위순회, 루트를 왼쪽과 오른쪽 서브트리 중간에 방문하면 중위순회, 루트를 서브트리 방문 후에 방문하면 후위순회가 된다.

여기서 항상 왼쪽 서브트리를 오른쪽 서브트리에 앞서서 방문한다고 가정한다. 다음 수식을 이진트리로 나타내면 아래 그림과 같다.

수식 : ((x - 3) + (y * z)) / 5



(가) 전위 표기법과 (나) 후위 표기법으로 올바르게 표현한 것은?(2006 교원)

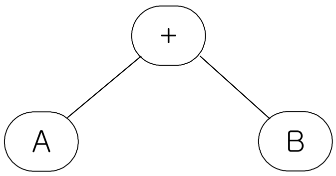
	(가) 전위 표기법	(나) 후위 표기법
①	x 3 - y z * + 5 /	/+ - x 3 * y z 5
②	/+ 5 - * x 3 y z	x - 3 + y * z + /5
③	x - 3 + y * z + /5	/+ 5 - * x 3 y z
④	/+ - x 3 * y z 5	x 3 - y z * + 5 /
⑤	/+ 5 - * x 3 y z	x 3 y z - * + 5 /

<풀이> ④

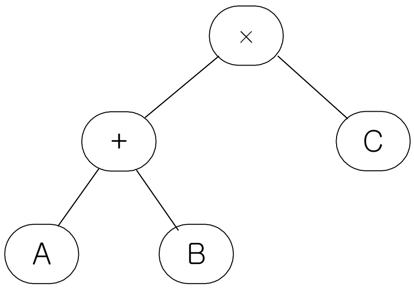
전위표기는 /+-x3*yz5이고 후위표기는 x3-yz*+5/이다.

문제Ⅲ.5.14

아래와 같이 이진트리(binary tree)를 사용하면 괄호와 사칙연산으로 이루어진 수식을 표현할 수 있다.(2007 중고등)

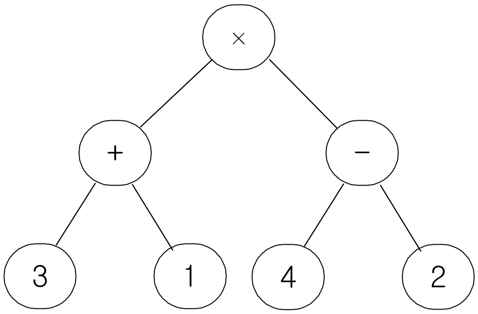


< A+B를 표현한 경우>



< (A+B)×C를 표현한 경우>

그렇다면 다음 이진트리가 표현하는 수식을 계산한 결과는 얼마인가?



① 5 ② 8 ③ 10 ④ 12 ⑤ 14

<**풀이**> ②

전위순회의 결과는 $\times + 31 - 42 - \times 42 = 8$이고,
후위순회의 결과도 $31 + 42 - \times = 42 \times = 8$이다.

마. 최소비용신장트리

그래프 G에서 G의 모든 정점들을 포함하는 트리를 G의 신장트리라고 한다. 깊이우선탐색을 이용하여 구성된 신장트리를 깊이우선신장트리, 너비우선탐색을 이용하여 구성된 신장트리를 너비우선신장트리라고 한다.

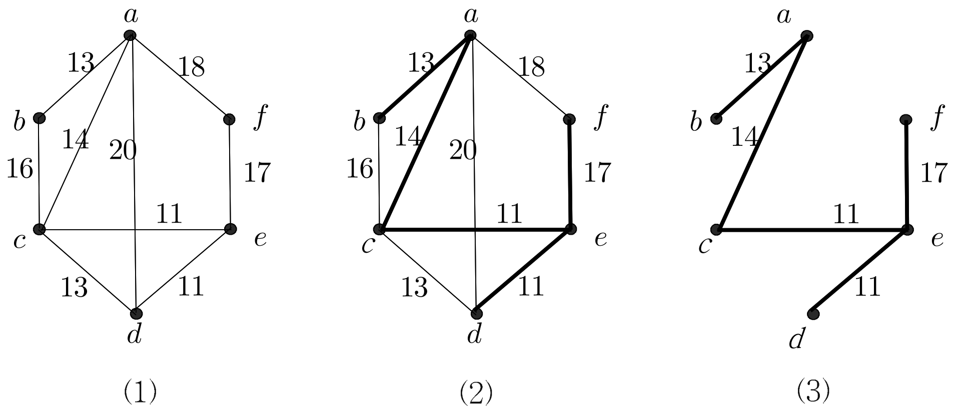
최소비용신장트리란 그래프 G의 신장트리 중에서 간선들의 비용의 합이 가장 작은 신장트리이다. 어떤 그래프의 최소비용신장트리는 유일하게 결정되는 것만은 아니고 비용의 합이 같은 다른 경로가 있을 수 있다. 최소비용신장트리를 구성하려면 그래프 G안에 있는 간선들만 사용해야 한다. 신장트리도 트리이므로 노드의 수가 n이라면 $n-1$개의 간선만 사용하고 순환이 되는 간선은 제외한다. 최소비용신장트리를 찾기 위해 보통 사용하는 알고리즘에는 Prim의 알고리즘과 Kruskal의 알고리즘이 있다.

1) Prim의 알고리즘

① 그래프 G에서 임의의 한 노드 a를 선택하여 노드 집합 A에 넣는다.
② a를 제외한 모든 노드 n에 대해 A에 있는 노드와 n사이의 최단 거리를 계산한 다음, A에 포함되어 있지 않고 거리가 최소가 되는 노드 n을 A에 추가한다.
③ 모든 노드가 A에 추가될 때까지 단계 ②를 반복한다.

〔그림Ⅲ.5.14〕의 가중그래프 (1)에서 Prim의 알고리즘을 이용하여 최소비

용신장트리를 구해보자. 먼저 A에 a를 넣는다. a에서 다른 노드들까지의 최단 거리 노드는 b이므로 b를 A에 추가한다. $A=\{a,b\}$에서 A에 속하지 않는 다른 노드들 중에서 가중치가 14인 최단거리 노드는 c이다. $A=\{a,b,c\}$가 되고 A에 포함된 노드와 포함되지 않은 노드사이에 거리가 최소가 되는 노드를 찾으면 e이므로 A에 e를 추가한다. $A=\{a, b, c, e\}$에서 남은 노드까지 거리가 최소가 되는 노드를 찾으면, 가중치 11의 d를 찾을 수 있고 d를 A에 추가한다. 마지막으로 A에 포함되어 있는 노드에서 포함되어 있지 않은 노드까지의 최단거리를 찾으면 노드 f이므로 f를 A에 추가한다. 〔그림Ⅲ.5.14〕의 (3)은 Prim의 알고리즘을 이용하여 만든 최소비용신장트리이다.



[그림Ⅲ.5.14] Prim의 알고리즘으로 최소비용신장트리 형성

2) Kruskal의 알고리즘

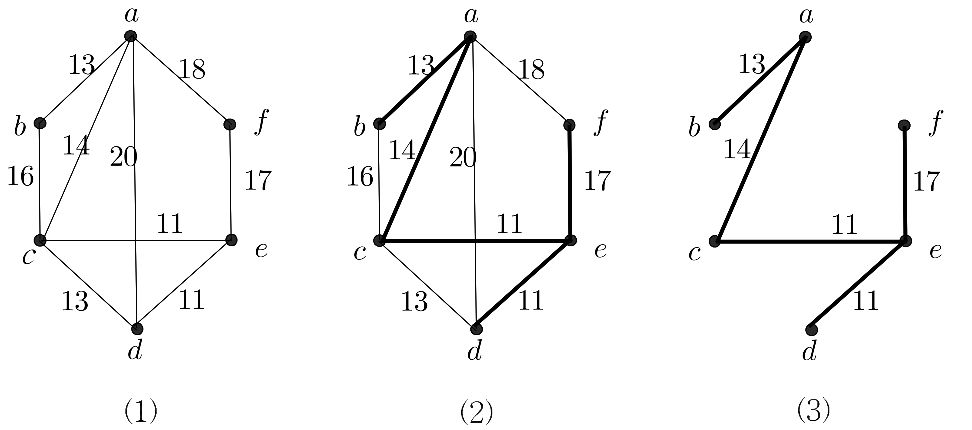
① 그래프 G에서 가장 작은 가중치를 갖는 간선을 선택한다.

② 그 다음으로 작은 가중치를 갖는 간선을 선택하면 된다. 그러나 선택된 간선이 순환이 되면 그 간선은 제외한다.

③ 모든 노드를 다 방문되면 최소비용신장트리가 구해진다.

〔그림Ⅲ.5.15〕의 가중그래프 (1)에서 Kruskal의 알고리즘을 이용하여 최소비용신장트리를 구해보자. 먼저 가장 작은 가중치를 갖는 간선을 선택한다. 간선 (c, e)를 선택한다. 그 다음으로 작은 가중치를 갖는 (d, e)를 선택하고 그 다음으로 작은 가중치를 갖는 간선이 2개가 있는데, 순환이 되는 간선은

제외해야 하므로 (a, b)를 선택한다. 계속해서 작은 가중치를 갖는 (a, c)와 (e, f)를 차례대로 선택하면 〔그림Ⅲ.5.15〕의 (3)와 같다.



[그림Ⅲ.5.15] Kruskal의 알고리즘으로 최소비용신장트리를 형성

Prim 알고리즘을 사용한 〔그림Ⅲ.5.14〕의 (3)과 Kruskal 알고리즘을 사용한 〔그림Ⅲ.5.15〕의 (3)은 동일한 것을 알 수 있다.

예제Ⅲ.5.15

아래 그림은 네 도시와 도시간 거리를 표현한 지도이다. 어느 한 외판원이 도시 A를 출발하여 모든 도시를 한번씩 방문하고 다시 도시 A로 돌아오려고 한다. 이동 거리가 최소가 되는 길을 선택한다고 할 때 그 최소 이동거리는? (2002중등)

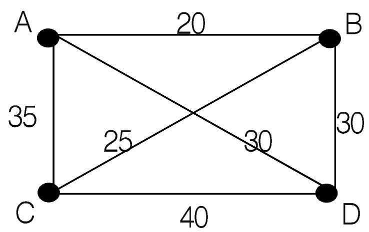
① 110

② 115

③ 125

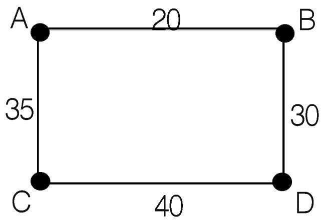
④ 140

⑤ 155

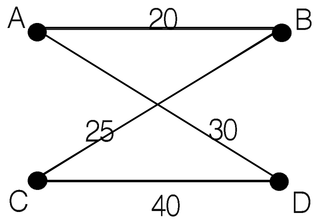


<풀이> ②

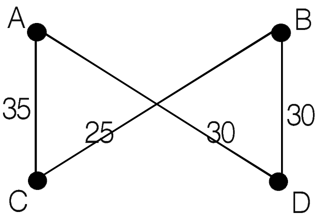
아래의 그림에서 ㉡의 최소이동거리가 115로 가장 작다.



㉠ 최소이동거리=125



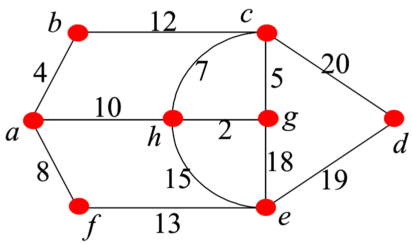
㉡ 최소이동거리=115



㉢ 최소이동거리=120

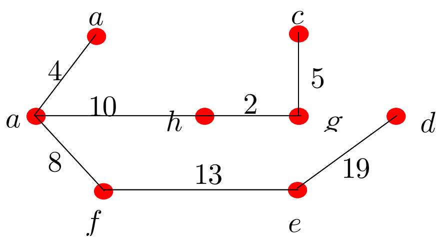
문제Ⅲ.5.15

아래 그림은 각 도시간 거리를 표현한 그림이다. 각 점은 도시를 나타내고 각 선은 도시간 거리를 나타낸다. 두 도시를 잇는 비용이 두 도시의 거리와 같다고 할 때 모든 도시들이 연결되도록 네트워크를 만드는 최소 비용은 얼마인가?(예를 들어 a도시와 b도시를 연결하는 데는 4만큼의 비용이 든다.)
(2002 고등)



<풀이> 61

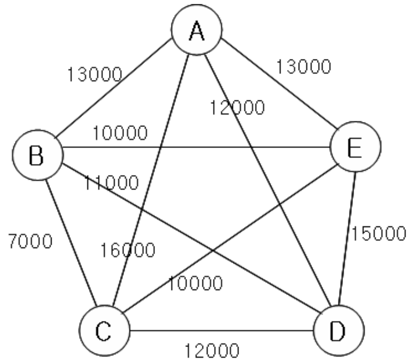
Prim의 알고리즘이나 Kruscal의 알고리즘을 이용하면 아래와 같은 최소가중비용트리를 얻을 수 있다.



예제Ⅲ.5.16

통신망, 도로망, 유통망 등은 대개 길이, 구축비용, 전송시간 등의 가중치를

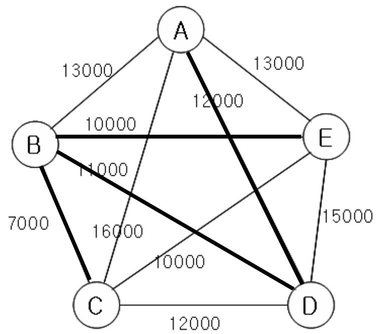
간선에 할당한 그래프인 네트워크로 표현될 수 있다. 이러한 도로망, 통신망, 유통망을 가장 적은 비용으로 구축하고자 한다면, 네트워크에 있는 모든 정점들을 가장 적은 수의 간선과 비용으로 연결하는 최소비용신장트리(minimum spanning tree)가 필요하게 된다. 즉, 신장트리의 비용은 트리에 포함된 모든 에지의 가중치를 합한 값이며, 최소비용신장트리는 비용이 최소가 되는 신장트리를 말한다. 아래 그림의 최소 비용은 얼마인가? (2006 교원)



① 60000 ② 47000 ③ 44000
④ 40000 ⑤ 33000

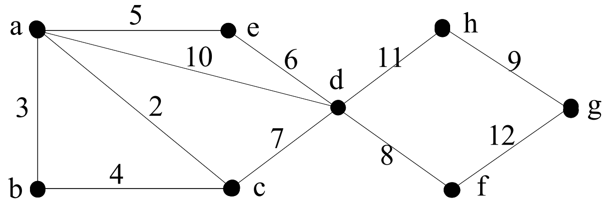
<풀이> ④

Prim의 알고리즘이나 Kruskal의 알고리즘을 이용하여 최소비용신장트리를 구하면 다음과 같다.



문제Ⅲ.5.16

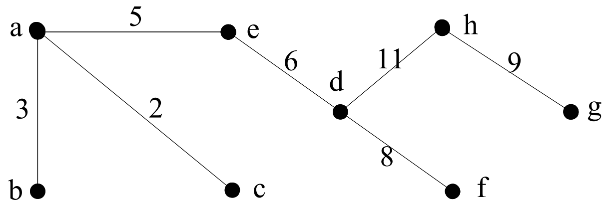
다음 가중그래프에서 모든 도시들이 연결되도록 네트워크를 만드는 최소비용은 얼마인가?



① 42 ② 43 ③ 44 ④ 45 ⑤ 46

<풀이> ③

최소비용신장트리를 구하면 다음과 같다.



바. 이진탐색트리

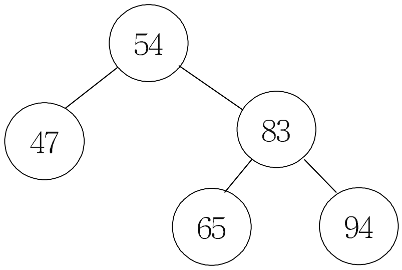
이진탐색트리(이진검색트리, binary search tree)는 루트의 값이 왼쪽 서브트리에 있는 임의의 노드의 값보다 크고 오른쪽 서브트리에 있는 임의의 노드의 값보다 작은 이진트리의 특수한 형태이다. 각각의 노드는 항목들 중의 하나인 키(key)에 의해 레벨을 부여받고 이 키는 노드를 식별하는데 사용되며 유일한 값을 가진다. 이진탐색트리는 특정 자료를 검색하거나 많은 자료를 순차적으로 정렬할 때 사용된다.

1) 이진탐색트리

이진탐색트리는 공집합이 가능하지만 공집합이 아닌 이진탐색트리는 다음 성질을 만족한다.

① 모든 노드는 키(key) 값을 가지며, 유일하다.
② 왼쪽 서브트리에 있는 키들은 그 서브트리의 루트보다 작아야 한다.
③ 오른쪽 서브트리에 있는 키들은 그 서브트리의 루트보다 커야 한다.
④ 왼쪽 서브트리와 오른쪽 서브트리도 모두 이진탐색트리이다.

이진탐색트리에서는 루트보다 큰 키 값을 갖는 노드들은 루트의 오른쪽 서브트리를 구성하고 작은 키 값을 갖는 노드들은 루트의 왼쪽 서브트리를 구성하게 된다.
〔그림Ⅲ.5.16〕은 이진탐색트리의 예이다.



[그림Ⅲ.5.16] 이진탐색트리

2) 이진탐색트리의 연산

가) 탐색

탐색(search) 연산은 이진탐색트리에서 많이 사용되는 연산이다. 탐색은 찾고자 하는 노드를 찾을 때까지 왼쪽 서브트리나 오른쪽 서브트리를 다음과 같은 과정으로 탐색하는 것이다.

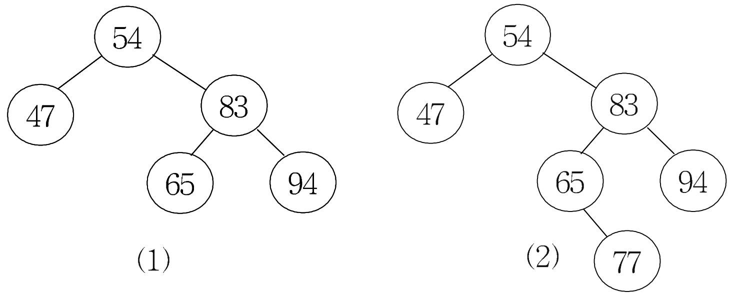
① 찾고자 하는 노드 값을 루트와 비교해서 루트보다 값이 크면 오른쪽 서브트리를 탐색하고, 루트보다 값이 작으면 왼쪽 서브트리를 탐색한다. ② 각 서브트리에 대해서도 단계 ①을 순환적으로 적용한다.

〔그림Ⅲ.5.16〕에서 65를 찾는 과정을 살펴보자. 먼저 65와 트리의 루트 54와 비교한다. 찾고자 하는 값이 더 크므로 오른쪽 서브트리로 가서 오른쪽 서브트리의 루트 83과 비교한다. 65는 83보다 작으므로 왼쪽 자식노드 65와 비교하게 된다. 이 경우 찾고자 하는 자료와 일치하므로 탐색을 종료하고 65의 포인터를 반환한다. 만약 탐색을 실패하게 되면 해당 자료가 이진탐색트리에 없으므로 NULL값을 반환하게 된다.

나) 삽입

이진탐색트리에 노드를 삽입(insert)하는 과정에서 노드가 삽입되는 위치는 이진탐색트리의 성질과 노드의 삽입 순서에 따라 유일하게 결정된다. 똑같은 자료라고 하더라도 삽입되는 순서에 따라 이진탐색트리의 모양은 달라진다. 새로운 노드가 삽입되는 위치는 단말노드가 된다. 트리에 N값과 동일한 값이 없다고 가정하고 N값을 루트와 비교하여 크면 오른쪽 서브트리로 이동하고,

작으면 왼쪽 서브트리로 이동하는 순환 과정을 거치게 된다. 예를 들어 〔그림Ⅲ.5.17〕의 이진탐색트리 (1)에 77값을 삽입한다고 하자. 77이 트리의 루트보다 크므로 오른쪽 서브트리로 이동하고 오른쪽 서브트리의 루트 83과 비교하면 작으므로 노드 83의 왼쪽 서브트리로 이동한다. 77이 65보다 크므로 노드 65의 오른쪽 자식노드로 삽입한다. 〔그림Ⅲ.5.17〕의 (2)는 (1)에서 노드 77을 삽입한 후의 이진탐색트리의 모습이다.



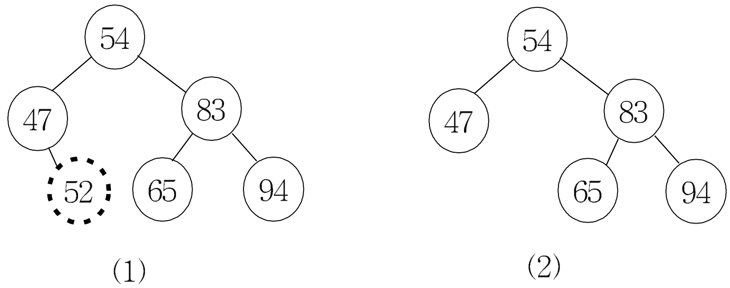
[그림Ⅲ.5.17] 노드 77을 삽입한 후 이진탐색트리

다) 삭제

이진탐색트리에서 노드의 삭제(delete)는 단말노드의 삭제, 자식이 하나인 노드의 삭제와 자식이 두 개인 노드의 삭제의 세 경우로 나누어 볼 수 있다.

(1) 단말노드의 삭제

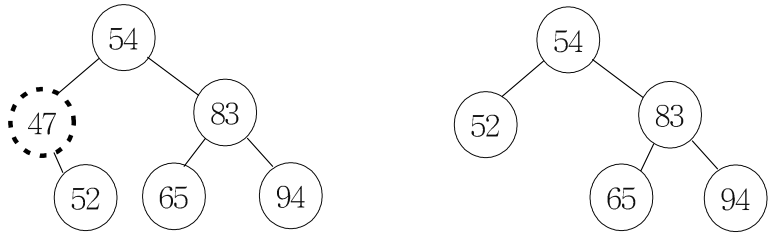
단말노드의 삭제는 해당 단말노드를 삭제하고 부모노드에서 그 단말노드를 가리키는 포인터를 NULL로 만든다. 〔그림Ⅲ.5.18〕의 (1)에서 단말노드 52를 삭제하려면 52의 부모노드인 47의 오른쪽 자식 필드를 NULL로 만든다.



[그림Ⅲ.5.18] 단말노드의 삭제

(2) 자식이 한 개인 단말노드의 삭제

자식을 하나만 가진 노드의 삭제는 그 노드를 삭제하고 자식노드를 삭제된 부모노드로 위치시키면 된다.

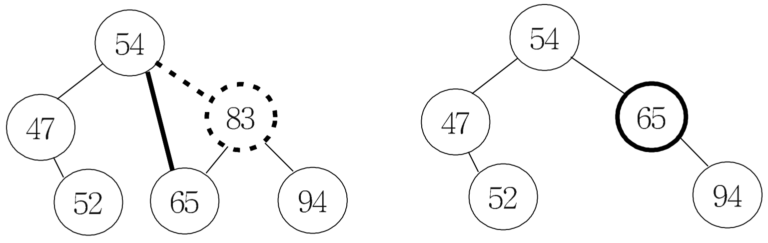


(1) (2)

[그림Ⅲ.5.19] 자식이 하나인 비단말노드의 삭제

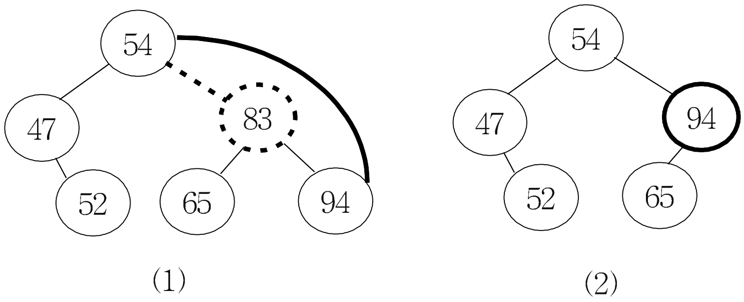
(3) 자식이 두 개인 비단말노드의 삭제

자식이 두 개인 비단말노드를 삭제하려면 해당 노드를 삭제하고 대신에 왼쪽 서브트리에서 가장 큰 노드나 오른쪽 서브트리에서 가장 작은 노도를 삭제된 노드의 위치로 대체하고 이진탐색트리를 형성하도록 순환적으로 과정을 진행한다. 〔그림Ⅲ.5.20〕에서 노드 83을 삭제한다고 하면 두 가지 경우가 가능하다. 먼저 〔그림Ⅲ.5.20〕의 (2)처럼 왼쪽 자식노드를 오른쪽 서브트리의 루트로 이동한다. 다른 방법은 먼저 〔그림Ⅲ.5.21〕의 (2)처럼 오른쪽 자식노드를 오른쪽 서브트리의 루트로 이동시킬 수도 있다.



(1) (2)

[그림Ⅲ.5.20] 두 개의 자식을 가진 비단말노드의 삭제



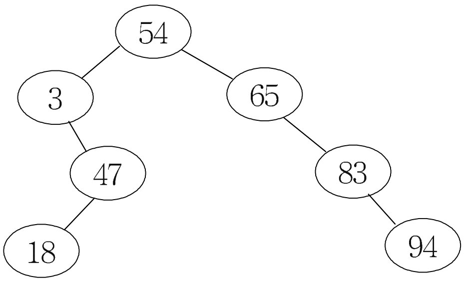
[그림Ⅲ.5.21] 두 개의 자식을 가진 비단말노드의 삭제

예제Ⅲ.5.17

자료 54, 3, 47, 18, 65, 83, 94를 가지고 이진탐색트리를 구성하시오.

<풀이>

① 54를 루트의 키로 둔다.
② 3은 54보다 작으므로 54의 왼쪽 자식노드에 배정한다.
③ 47은 54보다 작고 3보다 크므로 3의 오른쪽 자식노드에 배정한다.
④ 18은 47의 왼쪽 자식노드로 배정한다.
⑤ 65는 54보다 크므로 오른쪽 자식노드로 배정한다.
⑥ 83은 65보다 크므로 65의 오른쪽 자식노드에 배정한다.
⑦ 94는 83보다 크므로 83의 오른쪽 자식노드에 배정한다.



[그림Ⅲ.5.22] 이진탐색트리의 형성

문제Ⅲ.5.17

다음과 같이 정렬되어 있는 배열에서 이진탐색(binary search)으로 10의 위

치를 찾고자 한다. 10을 찾기까지 찾은 10을 포함해서 몇 개의 수를 10과 비교하는가? (2006 중고등)

1	3	4	7	10	12	14	17	23	31	37	42	45	57	63

① 3개　　② 4개　　③ 5개　　④ 6개　　⑤ 7개

<풀이> ②

min=1, max=15로 두고

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_i	1	3	4	7	10	12	14	17	23	31	37	42	45	57	63

① $\left\lfloor \frac{1+15}{2} \right\rfloor = 8$, 10은 $x_8 = 17$보다 작으므로 min=1, max=7

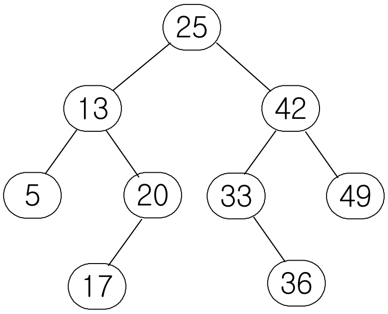
② $\left\lfloor \frac{1+7}{2} \right\rfloor = 4$, 10은 $x_4 = 7$보다 크므로 min=5, max=7

③ $\left\lfloor \frac{5+7}{2} \right\rfloor = 6$, 10은 $x_6 = 12$보다 작으므로 min=5, max=5

④ $\left\lfloor \frac{5+5}{2} \right\rfloor = 5$, 10은 $x_5 = 10$과 같다.

예제Ⅲ.5.18

다음과 같은 이진 탐색 트리(binary search tree)에 30을 추가하려고 한다. 30이 들어갈 위치로 알맞은 것은?(2007 초등)



① 33의 왼쪽 자식　　② 36의 왼쪽 자식　　③ 20의 오른쪽 자식
④ 49의 오른쪽 자식　　⑤ 17의 오른쪽 자식

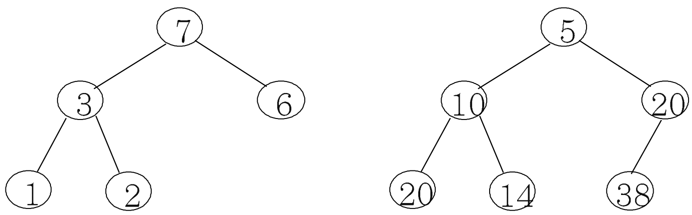
<풀이> ①
루트 노드 25와 비교하면 30이 더 크므로 오른쪽 자식 42와 비교한다. 30은 42보다 작으므로 42의 왼쪽 자식 33과 비교하고 더 작으므로 33의 왼쪽 자식 노드에 둔다.

문제Ⅲ.5.18
예제Ⅲ.5.18의 이진탐색트리에 18을 추가하려고 한다. 18이 들어갈 위치로 알맞은 것은?
① 33의 왼쪽 자식　　② 36의 왼쪽 자식　　③ 20의 오른쪽 자식
④ 49의 오른쪽 자식　　⑤ 17의 오른쪽 자식

<풀이> ⑤

사. 힙

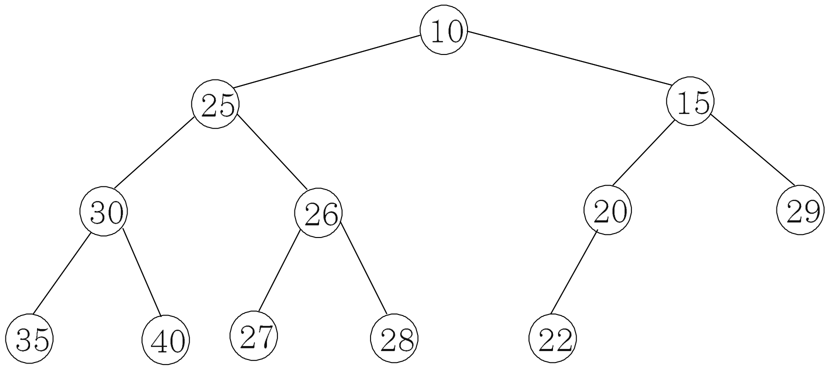
모든 부모 노드가 반드시 두 개의 자식 노드를 가지는 완전이진트리에서, 어떤 부모 노드를 선택해도 부모노드의 값이 자식노드의 값보다 작은 조건을 만족하는 트리를 최소 힙(mimimum heap)이라 한다. 최대 힙(maximum heap)은 각 노드의 값이 자식 노드의 값보다 크거나 같은 완전이진트리이다. 최소힙에서는 루트 노드의 값이 가장 작고 최대 힙에서는 루트의 값이 가장 크다. 힙에서는 왼쪽 자식 노드의 값과 오른쪽 자식 노드의 값 사이의 대소관계는 고려하지 않는다. 〈그림Ⅲ.5.23〉는 최소 힙과 최대 힙의 예이다.



〈그림Ⅲ.5.23〉 최대 힙과 최소 힙

힙 구조는 수시로 자료가 삽입되는 구조에서 삽입된 자료들 중에서 가장 큰 값(혹은 작은 값)을 갖는 자료를 찾는 경우에 사용한다. 예를 들어, 100만개의 자료 중에서 가장 큰 값(혹은 작은 값)을 찾을 경우 평균 50만번을 비교하게 된다. 이것을 힙 구조로 처리한다면 100만건의 자료가 있을 경우, 최악의 경우 약 20번, 평균적으로 10번의 비교로 검색을 마칠 수 있다.

힙을 1차원 배열로 나타내면 루트부터 같은 레벨에 있는 노드들을 왼쪽부터 오른쪽으로 차례대로 저장한다. 〈그림Ⅲ.5.25〉은 〈그림Ⅲ.5.24〉의 최소 힙을 1차원배열로 나타낸 것이다.



〈그림 Ⅲ.5.24〉 최소 힙

1	2	3	4	5	6	7	8	9	10	11	12
10	25	15	30	26	20	29	35	40	27	28	22

〈그림 Ⅲ.5.25〉 최소 힙을 배열에 저장

n개의 노드로 이루어진 힙을 1차원 배열로 나타내고, 자신이 저장된 위치의 첨자를 i라고 하면 다음과 같은 성질을 가진다.

① $i \neq 1$일 때, 부모 노드의 위치는 $\left\lfloor \frac{i}{2} \right\rfloor$ 이다.

② $2i \leqq n$일 때, 왼쪽 자식 노드의 위치는 $2i$이다.

③ $2i+1 \leqq n$일 때, 오른쪽 자식 노드의 위치는 $2i+1$이다.

새로운 자료를 최소 힙에 추가할 경우에는 빈 힙에서 시작하여 다음과 같은 과정을 수행한다.

① 새로운 자료를 힙의 마지막 요소로 저장한다.
② 저장한 요소를 자식 노드의 값으로 두고 이것을 부모 노드의 값과 비교해서 부모 노드의 값이 더 크면 자식 노드의 값과 부모 노드의 값을 교환한다.
③ 그런 다음 부모 노드를 자식 노드로 간주하고, 그 위에 있는 부모 노드에 대해 위 과정을 반복한다. 자식 노드의 위치가 루트 노드까지 도달했거나, 부모 노드의 값이 자식 노드의 값보다 작거나 같을 때까지 위 과정을 반복한다.

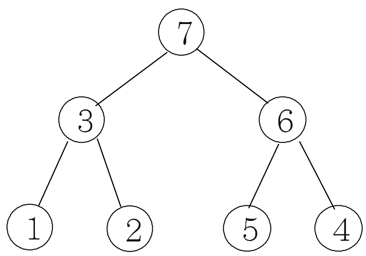
최대 힙에서 자료를 삭제하는 과정은 다음과 같다.

① 루트 노드를 삭제한다.
② 가장 마지막 노드의 값을 루트 노드에 두고 왼쪽 자식 노드와 오른쪽 자식 노드의 값과 비교해서 가장 큰 값을 루트 노드에 둔다.
③ 그런 다음 바뀐 노드를 부모 노드로 간주하고, 그 아래에 있는 자식 노드에 대해 위 과정을 반복한다. 더 이상 비교대상이 없을 때까지 위 과정을 반복한다.

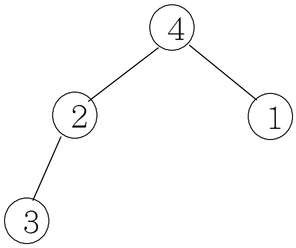
이와 같이 힙에서의 삭제는 루트 노드를 자른 다음 계속 힙을 만들어야 한다. 힙은 완전이진트리가 되어야 하므로 마지막 노드를 자른 다음 세 값을 비교 후 가장 큰 자료가 올라오는 동작을 반복하다가 자른 자료가 가장 크거나 같으면 해당 자리에 위치시킨다.

예제Ⅲ.5.19

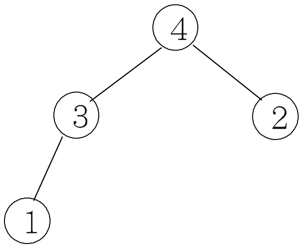
다음과 같은 힙(heap)에서 가장 일반적인 방법으로 세 개의 원소를 삭제하면 힙의 모양은 어떻게 되는가? (2005 중고등)



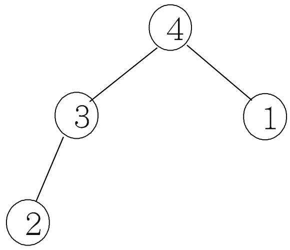
①



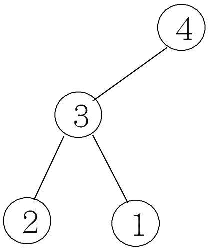
②



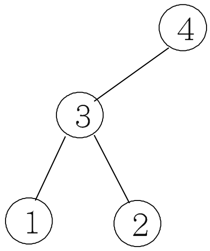
③



④

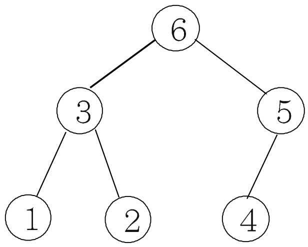


⑤

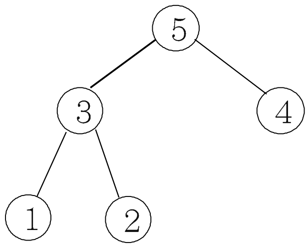


<풀이> ②

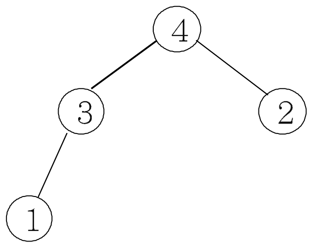
단계 ①에서 먼저 루트 노드의 7을 삭제한다. 가장 마지막 노드의 4, 3, 6을 비교한 후 가장 큰 값 6이 루트 노드에 위치한다. 단계 ②에서는 루트 노드 6을 삭제한다. 가장 마지막 노드에 있는 5, 3, 4를 비교한 후 루트 노드에 5를 둔다. 마지막으로 단계 ③에서 루트 노드의 5를 삭제하고 가장 마지막 노드의 2와 3, 4를 비교하면 4가 가장 크므로 4를 루트 노드에 2를 4의 오른쪽 자식 노드에 위치시킨다.



단계 ①



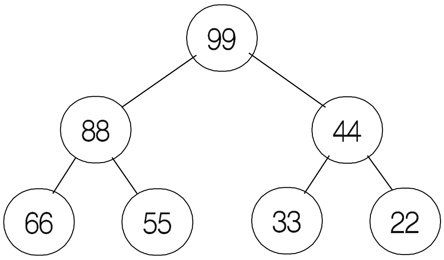
단계 ②



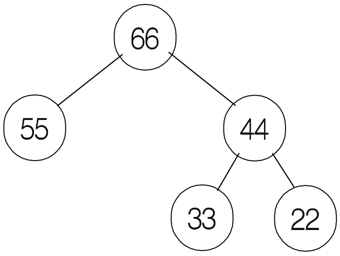
단계 ③

문제Ⅲ.5.19

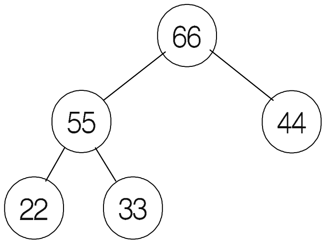
다음 최대 힙에서 두 개의 자료를 삭제(Delete)한 후 힙의 모습으로 알맞은 것은?



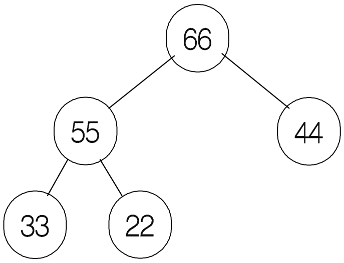
①



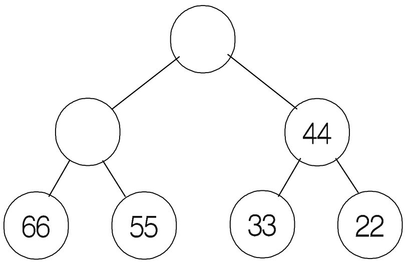
②



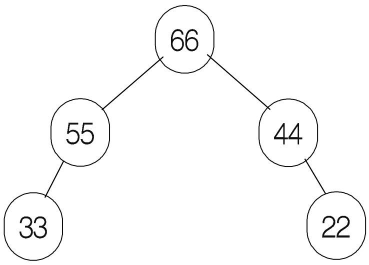
③



④

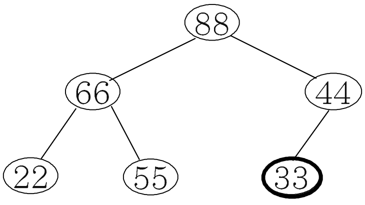


⑤

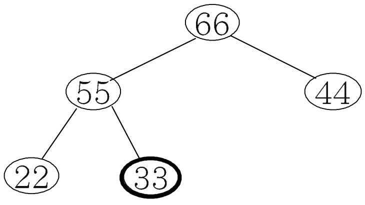


<풀이> ②

최대 힙에서는 루트 노드의 값이 가장 크다. 최대 힙에서의 삭제는 루트 노드에 있는 자료를 삭제하고 가장 마지막 노드를 루트 노드로 옮긴 다음 힙으로 만들면 다음과 같다.



1번째 원소를 삭제 후



2번째 원소를 삭제 후

아. 허프만 코드

문자 자료는 일반적으로 대소문자를 포함하는 영문 알파벳, 구두점 기호들과 !와 @ 등의 키보드 기호들로 구성된다. 컴퓨터는 이러한 문자 자료를 0과 1로 이루어진 이진 코드 형태로 저장한다. 가장 간단한 방법으로 이진 코드의 길이를 n으로 고정하면 모두 2^n개의 서로 다른 문자들을 표현할 수 있고 각 문자는 n개의 일련의 비트로 인코드된다. 인코드된 이진 코드를 디코드하면 원래의 문자로 나타난다. 인코드하는 방법 중에서 아스키코드가 많이 사용되고 있다.

아스키코드를 사용하면 사용빈도에 관계없이 모든 문자에 대해 같은 용량의 저장 공간을 필요로 한다. 각 문자에 대해 같은 수의 비트를 할당하는 것보다 다른 수의 비트들을 할당하는 가변 길이의 코드 방식을 사용하면 저장 공간의 용량을 줄일 수 있다. 발생빈도가 높은 문자들은 적은 비트를 할당하고, 발생빈도가 낮은 문자에는 그보다 많은 비트를 할당하여 저장 공간의 용량을 줄이는 것이다. 이를 자료 압축이라고 하는데 기본적인 방법으로 허프만 코드가 있다.

예제Ⅲ.5.20

어떤 파일의 자료가 전체 100개의 l, m, n, o, p라는 5개의 문자들로 구성되어 있으며 각 문자의 빈도수(%)는 40, 15, 25, 7, 13이라고 하자.

(1) 고정 길이의 비트 스트링으로 표현할 경우 각 문자에 대해 최소 몇 개의 비트가 필요한가? 이 때 필요한 저장공간은 얼마인가?

(2) 가정 길이의 비트 스트링으로 표현하여 0, 110, 10, 1110, 1111로 나타낸다고 할 때, 저장 공간은 얼마나 필요한가?

<풀이>

(1) 5개의 문자를 고정 길이의 비트 스트링으로 표현할 경우에는 3개의 비트가 필요하고($2^2=4<6<8=2^3$) 저장 공간은 $100\times3=300$비트가 필요하다.

(2) 가변 길이의 비트 스트링으로 표현하여 각각 0, 110, 10, 1110, 1111로 나타낸다고 하면,

$100\times(0.40\times1+0.15\times3+0.25\times2+0.07\times4+0.13\times4)=215$비트가 필요하게 된다.

허프만은 문자들의 발생 빈도로부터 허프만 코드를 구성하는 알고리즘을 제시하였다. 허프만 알고리즘은 다음과 같다.

① 빈도수가 적은 문자에서 큰 문자 순으로 나열한다. ② 가장 빈도수가 낮은 두 문자를 선택하여 새로운 노드를 만든다. 새로운 노드의 빈도수는 두 문자의 빈도수의 합으로 나타낸다. 이와 같은 과정을 모든 문자의 빈도수의 합으로 나타내는 하나의 이진트리가 완성될 때까지 반복한다. ③ 트리가 완성되면 루프노드에서 단말노드까지 왼쪽 서브 트리에는 0을, 오른쪽 서브 트리에는 1을 부여하여 각 문자의 코드를 완성한다.

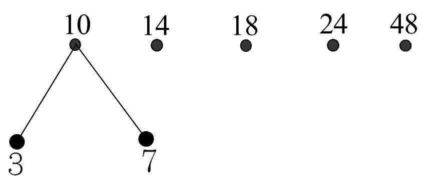
문제Ⅲ.5.20

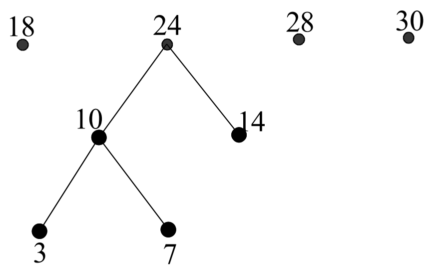
p. q, r, s, t, u의 빈도수(%)는 7, 3, 14, 18, 24, 48이라고 하자. 이에 대한 허프만 트리를 만들고 각 문자들의 허프만 코드를 구하여라.

<풀이>

① 주어진 문자를 빈도수에 대하여 오름차순으로 나열한다.

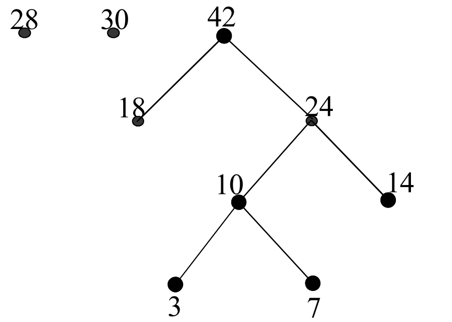
② 현 상태에서 빈도수가 가장 낮은 두 노드를 선택하여 새로운 노드를 만들어 오름차순이 유지되도록 노드를 삽입한다.



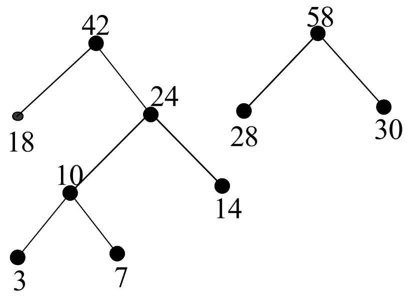


㉠ 노드 10을 만들고 3과 7은 10의 서브 노드로 이동

㉡ 노드 24를 만들고 10과 14는 24의 서브 노드로 이동

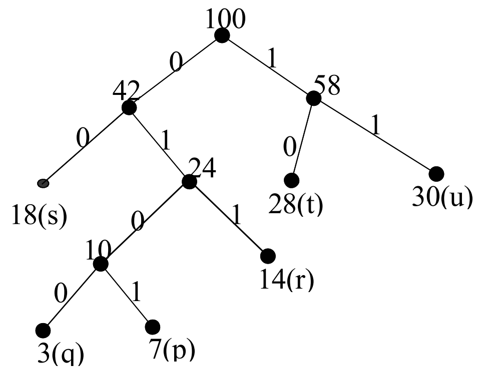


ㄷ 노드 42를 만들고 18과 24는 42의 서브 노드로 이동



ㄹ 노드 58을 만들고 28과 30은 52의 서브 노드로 이동

위와 같은 과정을 반복한 결과는 다음과 같다.

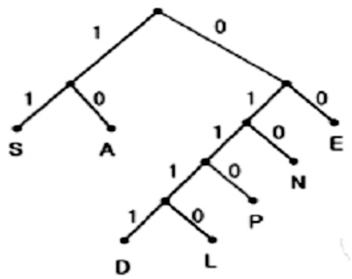


③ p, q, r, s, t, u에 대한 허프만 코드는 각각 0101, 0100, 011, 00, 10, 11이다.

예제Ⅲ.5.21

허프만 코드(Huffman code)는 문자들을 가변 길이의 비트 스트링으로 표현하는 것으로, ASCII나 다른 고정 길이 코드들에 대안을 제공한다. 기본 개념은 많이 사용되는 문자들을 표현하기 위해서는 짧은 길이의 비트 스트링을 사용하는 것이다. 허프만 코드는 트리에 의해 쉽게 정의된다. 비트 스트링을 디코드하기 위해서는, 뿌리에서 시작하여 문자를 만날 때까지 아래로 내려간다. 비트 0 또는 1은 오른쪽 또는 왼쪽으로 내려갈 것인지를 알려준다.

예를 들어 왼쪽의 그림과 같이 주어진 허프만 코드에서 A를 나타내는 비트 스트링은 10이고 S를 나타내는 비트스트링은 11이다. 다음 중 DEN을 나타내는 비트 스트링은 무엇인가?(2006 교원)



① 0111110010 ② 0111100010 ③ 0111000110
④ 0111110110 ⑤ 0111100110

<풀이> ②

D의 허프만 코드는 01111, E의 허프만 코드는 00, N의 허프만 코드는 010이다. 따라서 DEN의 허프만 코드는 011100010이다.

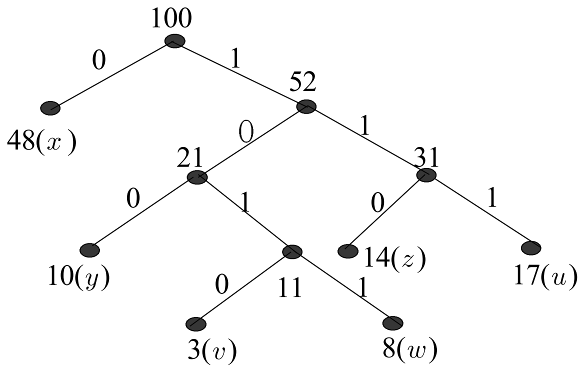
문제Ⅲ.5.21

다음 아스키문자 u, v, w, x, y, z에 대한 빈도수가 각각 17, 3, 8, 48, 10, 14일 때 각 문자들이 허프만 코드를 구한 것 중에서 잘못된 것은? (2007 교원)

① u 의 허프만 코드는 111이다.
② v 의 허프만 코드는 1010이다.
③ w 의 허프만 코드는 1011이다.
④ x 의 허프만 코드는 1이다.
⑤ y 의 허프만 코드는 100이다.

<풀이> ④

x의 허프만 코드는 0이다.

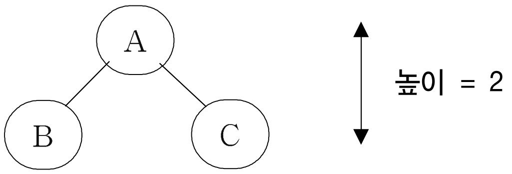


연 습 문 제

1. 트리 모양의 연결된(connected) 그래프가 10개의 정점(vertex)을 가질 때, 간선(edge)의 수는?

① 8개 ② 9개 ③ 10개 ④ 11개 ⑤ 20개

2. 다음 트리는 높이(height)가 2인 트리이다. 노드(node)의 개수가 1500개인 완전이진트리(complete binary tree)의 높이는 얼마인가?



① 9 ② 10 ③ 11 ④ 12 ⑤ 13

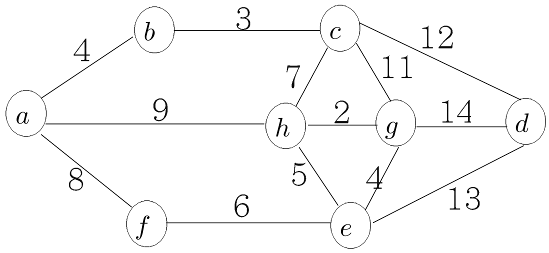
3. 이진트리에서 차수가 1인 노드 수가 15개이고 단말노드의 수가 20개이면 전체 노드의 수는 모두 몇 개인가?

① 53 ② 54 ③ 55 ④ 56 ⑤ 57

4. 어떤 이진트리를 전위순회(preorder traversal)한 결과가 ABDEHCFIG이고, 중위순회(inorder traversal)한 결과는 DBHEAFICG이었다.
이 이진트리를 후위순회(postorder traversal)한 결과가 올바른 것은?

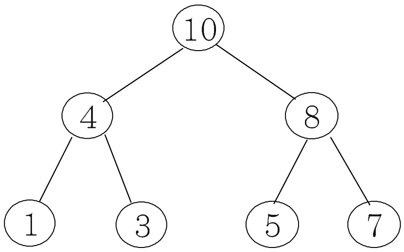
① ABCDEFGHIJ ② DHEBIFGCA ③ HDEBIFGCA
④ DEHBIFCGA ⑤ IHGFEDCBA

5. 아래 그림은 각 도시간 거리를 표현한 그림이다. 각 점은 도시를 나타내고 각 선은 도시간 거리를 나타낸다. 두 도시를 잇는 비용이 두 도시의 거리와 같다고 할 때 모든 도시들이 연결되도록 네트워크를 만드는 최소 비용은 얼마인가?(예를 들어 a도시와 b도시를 연결하는 데는 4만큼의 비용이 든다.)

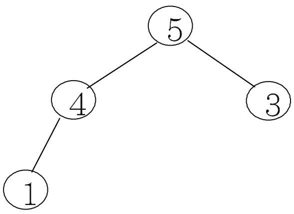


① 35　　② 36　　③ 38　　④ 45　　⑤ 45

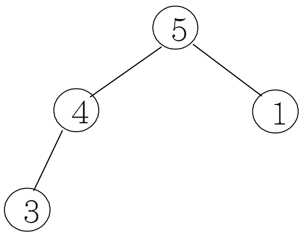
6. 다음과 같은 최대 힙(heap)에서 가장 일반적인 방법으로 세 개의 원소를 삭제하면 힙의 모양은 어떻게 되는가?



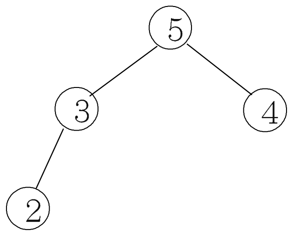
①



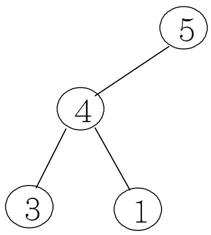
②



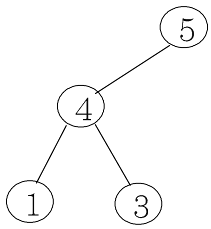
③



④



⑤



7. 다음과 같이 정렬되어 있는 배열에서 이진탐색(binary search)으로 10의 위치를 찾고자 한다. 10을 찾기까지 찾은 10을 포함해서 몇 개의 수를 10과 비교하는가? (2006 중고등)

2	4	6	8	10	11	13	16	17	18	19	20	24	26	30

① 3개 ② 4개 ③ 5개 ④ 6개 ⑤ 7개

8. 다음 수식을 이진트리로 나타내면 아래 그림과 같다.

수식 : (a + (2 / b)) * (a - b)

(가) 전위 표기법과 (나) 후위 표기법으로 올바르게 표현한 것은?

	(가) 전위 표기법	(나) 후위 표기법
①	* + a / 2 b - a b	a 2 b / + a b - *
②	* + a 2 b / a - b	a 2 b / + a - b *
③	a + * 2 - b a - b	a b - 2 b - a + *
④	a * + / 2 b a b	a 2 b a b + / - *
⑤	2 a b a b * + / -	2 a b a b - / + *

9. 다음의 문자와 빈도수에 대한 허프만 코드를 구한 것 중에서 잘못된 것은?

문자	a	i	n	r	t
빈도수	21	32	18	14	15

① a의 허프만 코드는 01이다.
② i의 허프만 코드는 10이다.
③ n의 허프만 코드는 00이다.
④ r의 허프만 코드는 100이다.
⑤ t의 허프만 코드는 101이다.

풀 이

1 ②

트리에서 간선의 수는 정점의 수보다 1이 작다.

2 ③

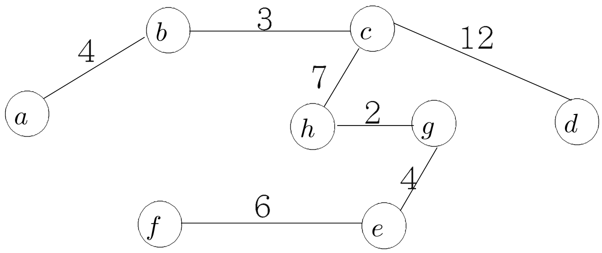
$2^{10}-1<1501<2^{11}-1$이므로 트리의 높이는 11이다.

3 ②

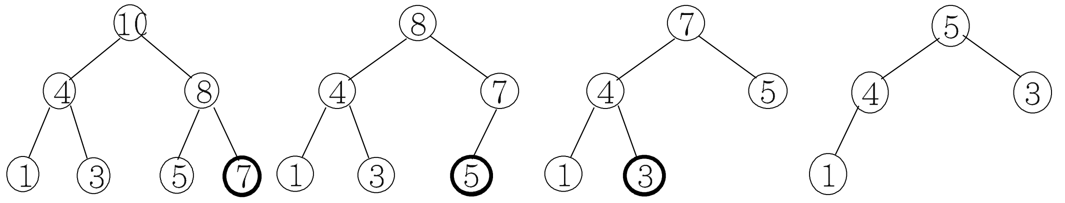
차수가 2인 노드 수는 단말노드의 수보다 1이 작으므로 19개가 된다. 따라서 전체노드수는 $20+15+19=54$이다.

4 ②

5 ③



6 ①



1번째 원소 삭제 후　2번째 원소 삭제 후　3번째 원소 삭제 후

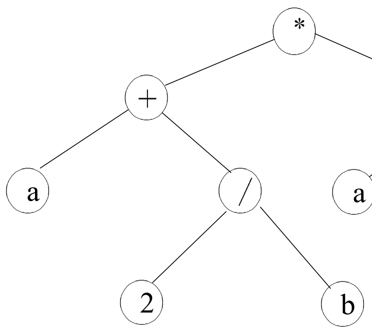
7 ②

min=1, max=15로 두고

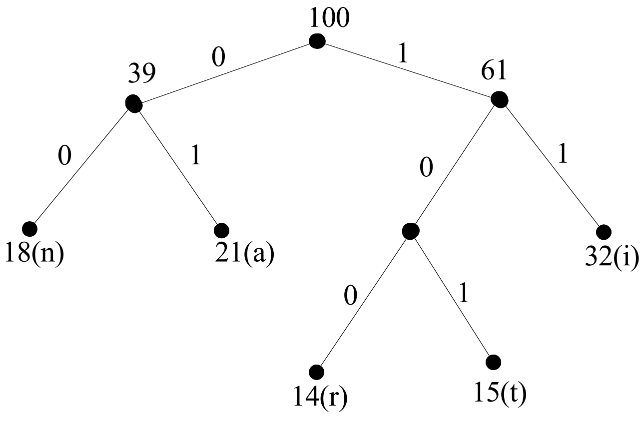
i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
x_i	2	4	6	8	10	11	13	16	17	18	19	20	24	26	30

8 ①

전위순회의 결과는 $*+a/2b-ab$이고 후위순회의 결과는 $a2b/+ab-*$이다.



9 ② i의 허프만 코드는 11이다.



6. 스택

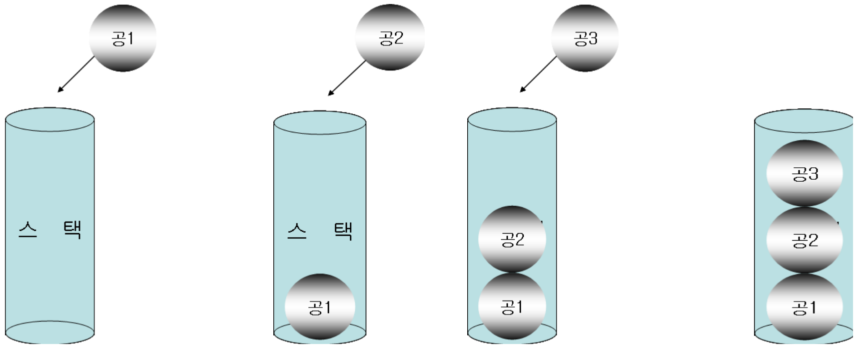
스택은 한 쪽은 막혀 있고 다른 쪽은 뚫려 있는 구조이다. 실제로 이런 구조가 존재하는 것이 아니라 이렇게 정의하여 사용을 한다. 한 쪽이 막혀 있는 구조이므로 다른 막히지 않은 쪽에서 자료의 삽입과 삭제가 발생한다. 일상생활에서 책장에 책을 쌓을 때나 아이들이 땅에서 일렬로 블록쌓기를 할 때 볼 수 있다.

가. 스택

스택은 대표적인 선형 자료 구조 중의 하나이다. 선형 자료 구조란 자료를 접근하는 방식이 순차적으로 이루어지는 자료 구조를 의미한다. 스택은 아래가 막힌 원통이라고 생각하면 이해하기 쉽다. 자료를 스택에 삽입하면 원통의 아래쪽부터 쌓이기 시작한다. 자료를 스택에서 삭제하게 되면 스택의 제일 위의 자료가 삭제된다. 이러한 자료 구조를 후입선출 구조(LIFO, Last In First Out)구조라고 한다. 그 이유는 제일 마지막에 삽입된 자료가 제일 먼저 삭제가 되기 때문이다. 스택에 자료를 삽입할 경우는 〈그림Ⅲ.6.2〉과 같이 위로부터 공을 넣을 수 있고, 아래는 막혀있다고 가정하자.

1) 스택에 자료를 삽입할 경우

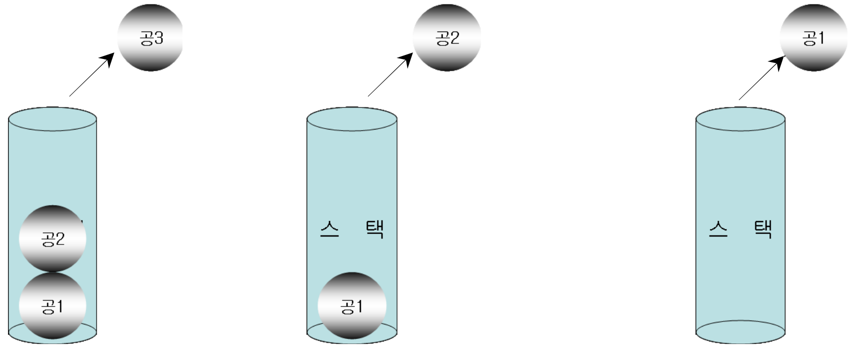
① 빈 스택에 공1을 넣는다.	② 스택에 공2, 공3을 차례대로 넣는다.	③ 세 공을 모두 넣은 상태는 아래와 같다.



〈그림Ⅲ.6.1〉 스택에 자료를 삽입하는 경우

스택에서 자료를 삭제할 경우는 〈그림Ⅲ.6.2〉과 같다.

① 스택에서 공3을 꺼낸다. ② 스택에서 공2를 꺼낸다. ③ 공3 - 공2 - 공1의 순으로 공이 나온다.



〈그림Ⅲ.6.2〉 스택에서 자료를 삭제하는 경우

스택의 연산은 〈그림Ⅲ.6.1〉과 〈그림Ⅲ.6.2〉처럼 기본적으로 삽입과 삭제가 있다. 삽입 연산은 스택에 새로운 자료를 하나 추가하는 것이고, 삭제 연산은 현재 스택에서 자료 하나를 제거하는 것이다. 일반적으로 스택에 자료를 삽입하는 연산을 Push라고 하고 자료를 삭제하는 연산을 Pop이라고 한다.

나. 스택의 구현

스택은 일반적으로 1차원 배열을 이용해서 구현할 수 있다. 자료를 스택에 쌓는 Push와 자료를 스택에서 추출하는 Pop 함수가 필요하다. 스택에서 자료가 어느 위치까지 저장되어 있는지를 관리하기 위해 스택의 삽입과 삭제가 일어나는 입구를 나타내는 스택 포인터가 필요하고 이 변수를 일반적으로 top 라고 한다. 자료가 스택에 Push될 때마다 top의 값은 1씩 증가하고 Pop할 때는 1씩 감소한다. 자료를 Push할 때 top의 값을 1씩 증가시키고 top이 가리키는 위치에 자료를 저장한다. 자료를 Pop할 때는 top이 가리키는 위치의 자료를 추출하고 top의 값을 1만큼 감소시킨다. 따라서 top이 초기값인 경우는 스택이 비어있는 상태이고 top이 스택의 최대크기인 경우는 가득 찬 상태이다. 여기서는 10진 정수를 100개 저장할 수 있는 스택을 만들어 보자.

스택의 선언

〈표Ⅲ.6.2〉 스택 배열의 선언

```
<VC>
int STACK[100];
                    //STACK라는 이름의 정수 배열 100개를 선언
```

```
<VB>
Dim STACK(100) As Integer
                    'STACK라는 이름의 정수 배열 100개를 선언
```

이 과정으로 100개를 저장할 수 있는 스택이 마련된다. 이 선언문을 보면 일반적인 배열의 선언과 차이점이 전혀 없다. 단지 사용하는 방법이 다를 뿐이다. 〈표Ⅲ.6.1〉의 스택에서 top의 값이 100이면 자료를 더 이상 넣을 공간이 없는 경우에 스택 오버플로우(overflow)가 발생하며, top의 값이 0이면 빼내올 자료가 없는 경우에 자료를 삭제하려는 경우에 스택 언더플로우(underflow)가 발생한다. 배열을 사용할 경우에는 100개의 공간 중에 원하는 어느 곳이라도 첨자를 사용해서 삽입과 삭제 연산을 할 수 있지만 배열을 스택으로 사용하면 정해진 연산 함수 Push와 Pop을 통해서만 삽입과 삭제 연산을 하게 되는 것이다.

스택 입구(TOP)의 선언

〈표Ⅲ.6.2〉 언어별 스택입구(top)의 선언

```
<VC>
int top = -1;
          //top은 스택의 입구를 나타내는 변수 초기 값은 -1
```

```
<VB>
Dim TOP As Integer
          'TOP은 스택의 입구를 나타내는 변수 초기 값은 0
          'Visual Basic의 경우는 선언 시 초기 값을 부여하지
          '않으면 자동으로 0이 초기 값이 된다.
```

우리는 스택을 사용할 때 top라는 변수를 통해서만 스택 배열에 접근할 수 있도록 해야 한다. 즉, top변수가 가리키는 곳이 스택의 입구가 된다. 배열로 구현한 스택을 그림으로 표현하면 다음과 같다.

〈표Ⅲ.6.3〉 언어별 스택의 초기 상태

<VC> 현재 스택의 값()

0	1	2	3	4	5	6	7	…	99	STACK

$\boxed{-1}$ top (현재 스택의 입구를 가리킴)

<VB> 현재 스택의 값()

1	2	3	4	5	6	7	8	…	100	STACK

$\boxed{0}$ top (현재 스택의 입구를 가리킴)

이 상태에서 10을 삽입하게 되면 스택은 다음과 같이 된다. 10을 삽입하는 연산은 Push(10)이라고 해야 한다. 즉 STACK〔0〕 = 10과 같은 연산은 사용할 수 없다.

〈표Ⅲ.6.4〉 스택에 하나의 DATA를 삽입한 상태

<VC> 현재 스택의 값(10)

10										
0	1	2	3	4	5	6	7	…	99	STACK

$\boxed{0}$ top (현재 스택의 입구를 가리킴)

<VB> 현재 스택의 값(10)

10										
1	2	3	4	5	6	7	8	…	100	STACK

$\boxed{1}$ top (현재 스택의 입구를 가리킴)

계속해서 3, 7, 2를 입력하게 되면 스택은 아래와 같은 상태가 된다.

〈표Ⅲ.6.5〉 스택에 4개의 값을 삽입한 상태

<VC> 현재 스택의 값(10, 3, 7, 2)

10	3	7	2						
0	1	2	3	4	5	6	7	…	99

STACK

3 top (현재 스택의 입구를 가리킴)

<VB> 현재 스택의 값(10, 3, 7, 2)

10	3	7	2						
1	2	3	4	5	6	7	8	…	100

STACK

4 top (현재 스택의 입구를 가리킴)

위의 그림에서 알 수 있듯이, top값은 스택의 입구를 가리키고 있고, 스택의 칸을 회색으로 표현하였다. 즉 입구에 자료 2가 있다는 의미로 해석하면 된다. 지금 이 상태에서 자료를 삭제하게 되면, 2가 제거될 것이고 스택의 입구는 한 칸 뒤로 이동해야 된다. 지금 이 상태에서 삭제 연산을 수행하면 결과는 그림과 같다.

〈표Ⅲ.6.6〉 스택에서 값을 삭제

<VC> 현재 스택의 값(10, 3, 7)

10	3	7	2						
0	1	2	3	4	5	6	7	…	99

STACK

2 top (현재 스택의 입구를 가리킴)

<VB> 현재 스택의 값(10, 3, 7)

10	3	7	2						
1	2	3	4	5	6	7	8	…	100

STACK

3 top (현재 스택의 입구를 가리킴)

위의 결과를 보면 2를 삭제했음에도 불구하고, 배열에서 2는 그대로 남아

있다. 하지만 2는 현재의 top의 값 외부에 위치하므로 스택에서 삭제되었다고 생각해도 된다. 스택에서 다시 삽입 연산을 수행하게 되면 5가 2를 덮어쓰게 된다. 위의 예에서 제시된 내용을 연산자로 표현해 보면 〈표Ⅲ.6.8〉과 같다.

〈표Ⅲ.6.7〉 스택에서 값을 삭제한 후 다시 삽입한 상태

<VC> 현재 스택의 값(10, 3, 7, 5)

10	3	7	5							
0	1	2	3	4	5	6	7	…	99	STACK

$\boxed{3}$ top (현재 스택의 입구를 가리킴)

<VB> 현재 스택의 값(10, 3, 7, 5)

10	3	7	5							
1	2	3	4	5	6	7	8	…	100	STACK

$\boxed{4}$ top (현재 스택의 입구를 가리킴)

〈표Ⅲ.6.8〉 스택 연산

<VC>

```
Push(10);
Push(3);
Push(7);
Push(2);
printf("%d",Pop());
Push(5);   // Push(a)는 a를 스택에 삽입, Pop()은 스택에서 값을 제거해서 반환
```

<VB>

```
Push 10
Push 3
Push 7
Push 2
Msgbox Pop()
Push 5        'Push a는 a를 스택에 삽입, Pop()은 스택에서 값을 제거해서 반환
```

예제Ⅲ.6.1

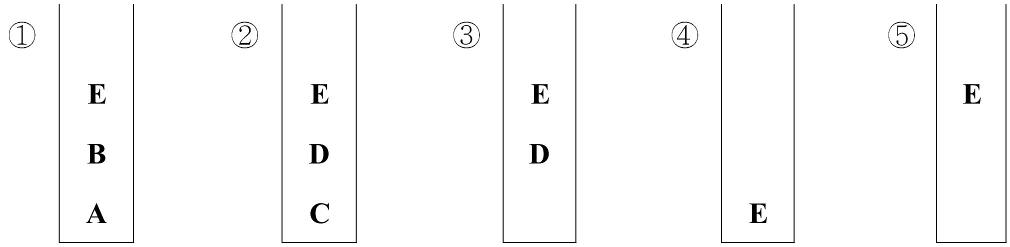
다음과 같은 네 가지 종류의 함수가 있다. 다음 프로그램을 차례로 실행시켰을 때 스택 s의 최종 상태는 어떻게 될까? (2002 중등 15번)

[VC++]
1. create(s) : 빈 스택을 s를 만든다.
2. empty(s) : 스택 s가 비어있으면 true, 그렇지 않으면 false를 돌려준다.
3. push(s, item) : 스택 s에 item을 삽입한다.
4. pop(s) : 스택 s에서 값을 하나 삭제한다.

```
create(s);
push(s, 'A');
push(s, 'B');
if (empty(s)) push(s, 'C'); else pop(s);
if (empty(s)) push(s, 'D'); else pop(s);
if (empty(s)) push(s, 'E'); else pop(s);
```

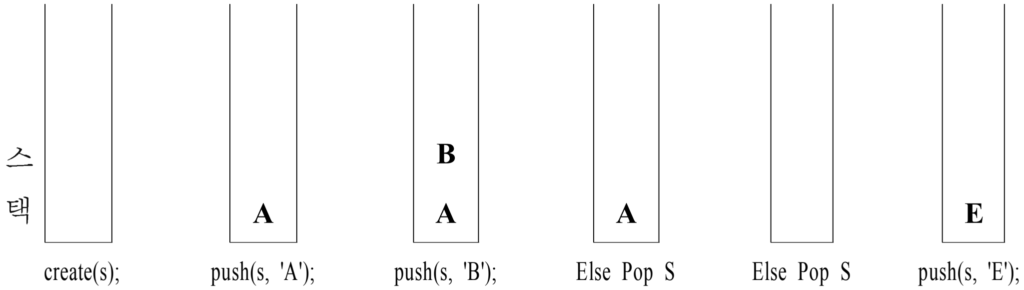
[VB]
1. Create S : 빈 스택을 S를 만든다.
2. IsEmpty(S) : 스택 S가 비어있으면 True, 그렇지 않으면 False를 돌려준다.
3. Push S, Item : 스택 S에 Item을 삽입한다.
4. Pop S : 스택 S에서 값을 하나 삭제한다.

```
Create S
Push S, "A"
Push S, "B"
If IsEmpty(S) Then Push S, "C" Else Pop S
If IsEmpty(S) Then Push S, "D" Else Pop S
If IsEmpty(S) Then Push S, "E" Else Pop S
```



<풀이> ④

프로그램을 실행시켰을 때 스택 s의 상태 변화는 다음과 같다.



문제Ⅲ.6.1

아래 프로그램은 스택(stack)을 사용한 프로그램이다. 사용된 함수의 기능은 다음과 같다.

Push(A) : A를 스택(stack)에 넣는다. GetValue( ) : 스택의 가장 위에 놓인 값이 무엇인지 알려 준다. Pop( ) : 스택에서 값을 하나 빼내어 돌려준다. IsEmpty() : 스택이 비어있으면 True(또는 true), 그렇지 않으면 False(또는 false)를 돌려준다.

그렇다면 아래 프로그램을 실행시켰을 때 출력되는 결과는 무엇인가? (2004 중고등)

< VC >	< VB >

<pre>#include <stdio.h>
void main()
{
    char s[6] = "KOREA";
    char r[6] = "";
    int   rn = 0;
 	
    int i;
 	
    for (i = 0; i < 5; i++)
    {
        if (!IsEmpty() && GeValuet() < s[i])
            r[rn++] = Pop();
        Push(s[i]);
    }
 	
    while (!IsEmpty())
        r[rn++] = Pop();
 	
    printf("%s\n", r);
}</pre>
	<pre>Sub main()
	
	    S = "KOREA"
	
	    For I = 1 To 5
	        If Not IsEmpty And
	           GetValue() < Mid(S, I, 1)
	           Then R = R + Pop()
	        Push (Mid(S, I, 1))
	    Next I
	
	    While Not IsEmpty()
	        R = R + Pop()
	    Wend
	
	    Debug.Print R
	
	End Sub</pre>

① KOREA ② AEROK ③ AEKOR ④ KOAER ⑤ ROKEA

<풀이> ④

배열 s, r이 있고 스택이 있다. 처음에는 빈 스택이므로 s[0]의 값인 K를 스택에 넣는다. 현재 스택이 비어있지 않고 스택의 가장 위에 놓인 값(K)과 s[1]의 값을 비교하면 K<O이므로 스택의 가장 위에 있는 K를 pop을 해서 K를 r[0]에 넣는다. i=1일 때, 스택의 가장 위에 놓인 값인 O보다 s[2]=R가 크므로 r[1]=O가 된다. 그러나 i=2일 때, 스택의 가장 위에 놓인 값인 R보다 s[3]=E는 크지 않으므로 배열 r에 값이 들어가지 않고 스택에 s[3]이 push된다. 마찬가지로 계속해서 스택에 R, E, A가 쌓이게 되고 아래의 while문에 의해 pop되어 배열 r에 A, E, R순으로 저장된다.

예제Ⅲ.6.2

다음 중 깊이 우선 탐색(DFS, depth-first search)과 너비 우선 탐색(BFS, breath-first search)을 할 때 일반적으로 사용하는 자료구조를 바르게 짝지은 것은?

① DFS - 스택(stack), BFS - 힙(heap)

② DFS - 스택(stack), BFS - 큐queue)

③ DFS - 큐(queue), BFS - 스택(stack)

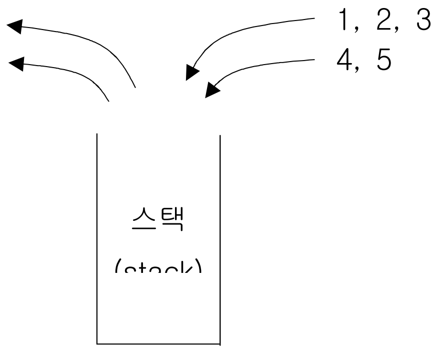
④ DFS - 큐(queue), BFS - 힙(heap)

⑤ DFS - 힙(heap), BFS - 스택(stack)

<풀이> ②

문제Ⅲ.6.2

1, 2, 3 세 개의 자연수를 차례대로 스택(stack)에 넣은 다음, 두 개의 자연수를 빼낸 뒤, 다시 두 개의 자연수 4, 5를 차례대로 스택에 넣고, 스택에 있는 모든 자연수를 빼내었다. 다음 중 자연수들이 스택에서 나온 순서로 알맞은 것은? (2005 초등 14번)

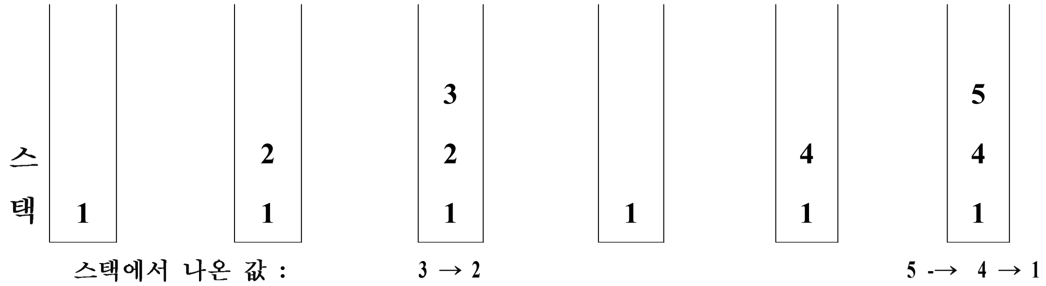


① 1, 2, 3, 4, 5 ② 1, 2, 4, 5, 3 ③ 2, 1, 5, 4, 3

④ 2, 3, 5, 4, 1 ⑤ 3, 2, 5, 4, 1

<풀이> ⑤

삽입과 삭제 후의 스택의 상태는 다음과 같다.



예제Ⅲ.6.3

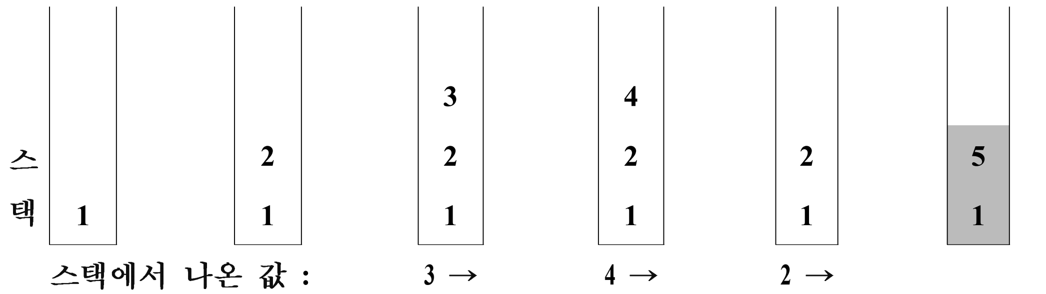
배열로 구현된 스택 A에 다음과 같이 삽입(push)과 삭제(pop)가 되풀이된 후 스택에 남아있는 값들의 합은 얼마인가? (2006 교원)

push(A,1) ⇒ push(A,2) ⇒ push(A,3) ⇒ pop(A) ⇒ push(A,4) ⇒ pop(A) ⇒ pop(A) ⇒ push(A,5)

① 3　② 4　③ 5　④ 6　⑤ 7

<풀이> ④

삽입과 삭제 후의 스택의 상태는 다음과 같다. 마지막으로 스택에 남아있는 값들의 합은 6이 된다.



연 습 문 제

1 다음의 빈칸에 적합한 단어는 무엇인가?

(　) 은 선형 자료구조중의 하나로 top이라고 하는 변수를 가지고 한 쪽 끝에서 자료의 삽입과 삭제가 일어나는 선형 자료 구조 중의 하나이다.

① 큐　② 트리　③ 스택　④ 그래프　⑤ 연결리스트

2 스택 알고리즘에서 top이 스택 포인터이고, m이 스택의 길이일 때, 서브루틴 “AA”가 처리해야 하는 것은?

```
top ← top+1
if top > m then goto AA
else STACK(T) ← item
```

① 오버플로우 처리　② 언더플로우 처리　③ 삭제 처리
④ 삽입 처리　⑤ 검색

3 다음 응용 분야에 이용되는 자료 구조는?

가. 인터럽트의 처리　나. 수식의 계산　다. 서브루틴의 복귀번지 저장

① 큐　② 트리　③ 스택　④ 그래프　⑤ 연결리스트

4 스택의 응용 분야와 거리가 먼 것을 다음 중에서 고르시오.

① 운영체제의 작업 스케줄링　② 함수 호출의 순서제어
③ 인터럽트의 처리　④ 수식의 계산
⑤ 서브루틴의 복귀번지 저장

5 배열로 구현된 스택 A에 다음과 같이 삽입(push)과 삭제(pop)가 되풀이된 후 스택에 남아있는 값들의 합은 얼마인가?

push(A,1) ⇒ pop(A) ⇒ push(A,2) ⇒ push(A,3) ⇒ pop(A) ⇒ push(A,4) ⇒ pop(A) ⇒ push(A,5) ⇒ push(A,6)

① 9　② 10　③ 12　④ 13　⑤ 17

풀 이

1 ③

2 ①

3 ③

4 ①

스택은 함수의 호출이나 인터럽트를 처리할 때 현재의 주소나 상태를 임시적으로 저장하는데 사용한다.

5 ④

삽입과 삭제 후의 스택의 상태는 다음과 같다. 마지막으로 스택에 남아있는 값들의 합은 13이 된다.

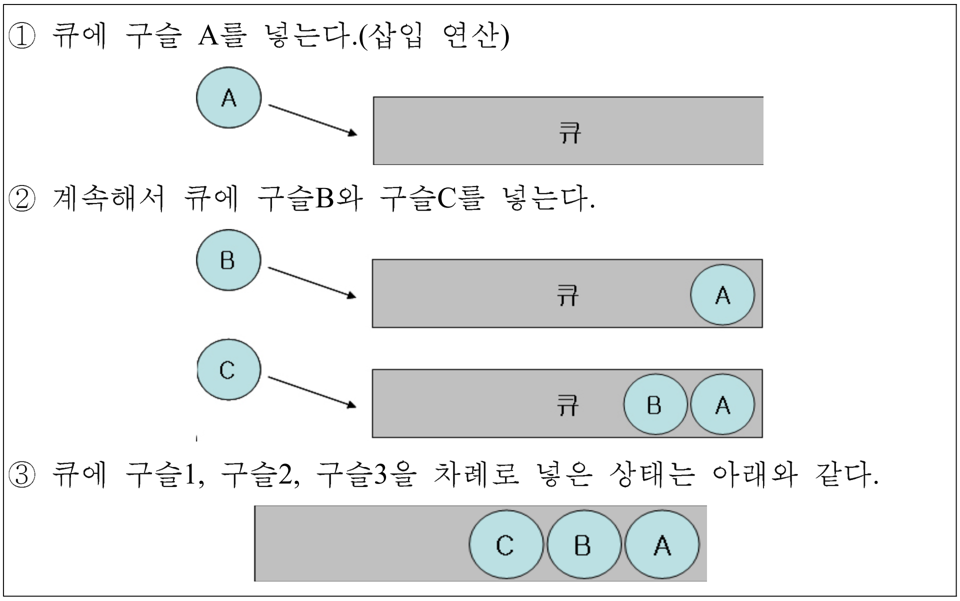
						6
스		3		4		5
택	1	2	2	2	2	2
스택에서 나온 값 :			1 →	3 →	4	

7. 큐

스택은 자료의 저장 순서와 반대로 자료를 추출하게 되는 방식이었다. 만약, 은행이나 화장실 등에서 대기행렬을 스택으로 처리하게 된다면 나중에 온 사람이 먼저 처리되므로 불공평하게 된다. 이 경우에 먼저 온 사람들이 먼저 처리되는 방식의 자료 구조가 필요한데, 이것을 큐라고 한다.

가. 큐

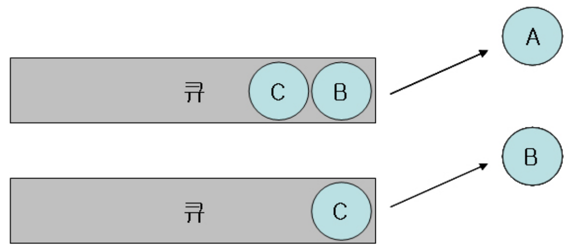
우리가 줄을 설 때는 기존 줄의 뒤에서 서고 앞에서부터 들어가게 된다. 이런 구조를 큐(Queue)라고 하고 큐란 영어로 줄을 서다라는 뜻이다. 큐는 스택과 마찬가지로 대표적인 선형 자료 구조 중의 하나이다. 스택은 한 개의 입구로 입력과 출력이 이루어진 반면, 큐는 입구와 출구가 따로 존재하는 자료 구조이다. 그렇기 때문에 큐는 먼저 입력된 자료가 먼저 출력되는 선입 선출(FIFO, First In First Out)구조이다. 큐는 양쪽 끝이 뚫려있는 긴 원통이라고 생각하면 된다. 긴 원통과 공을 가지고 구현한 큐를 살펴보도록 하자. 큐에 자료를 삽입하는 경우는 〈그림Ⅲ.7.1〉과 같다.



〈그림Ⅲ.7.1〉 큐에 자료를 삽입하는 경우

큐에 자료를 삭제하는 경우는 〈그림Ⅲ.7.2〉과 같다.

① 큐에서 구슬A를 꺼낸 다음, 구슬B를 꺼내 보자. (삭제 연산)



〈그림Ⅲ.7.2〉 큐에 자료를 삭제하는 경우

구슬이 나오는 순서는 구슬A - 구슬B - 구슬C의 순이다.

위의 예를 보면 큐의 구조와 연산을 이해할 수 있다. 큐의 연산은 삽입과 삭제가 있다. 삽입 연산은 큐에 새로운 자료를 넣는 작업이고, 삭제 연산은 현재 큐에서 제일 먼저 들어왔던 자료를 제거하는 작업이다. C언어에서 객체로 제공되는 큐에서는 삽입을 push라고 하고 자료를 삭제하는 연산을 pop이라고 하므로 이 교재에서도 큐의 삽입 삭제 연산을 스택과 동일한 이름으로 사용한다.

나. 큐의 구현

큐도 스택과 마찬가지로 1차원 배열로 구현할 수 있다. 큐를 구성하기 위해서 필요한 연산이 Push와 Pop이다. 큐에서는 두 개의 변수가 필요하다. 삽입할 경우 뒤에서 줄을 서야 하므로 뒤를 가리키는 변수 Rear(혹은 Tail)가 필요하고 삭제할 경우에는 앞에서부터 삭제되므로 앞을 의미하는 Front(혹은 Head)가 필요하다. 배열을 큐로 사용할 때 Rear와 Front 변수를 통해서만 큐에 접근한다. 즉 Rear와 Front 변수가 가리키는 곳이 큐의 출입구가 된다. 자료를 추출하고 저장할 때마다 Front와 Rear은 각각 1씩 더해지므로 언젠가는 배열의 마지막 요소에 도달한다. 〈표Ⅲ.7.1〉에서 배열을 이용하여 10진 정수를 100개 저장할 수 있는 큐를 만들어 보도록 하자.

큐의 선언

〈표Ⅲ.7.1〉 큐의 선언

<VC> int Queue[100];
<VB> Dim QUEUE(100) As Integer

이 과정으로 100개를 저장할 수 있는 큐가 만들어진다. 이 선언문을 보면 일반적인 배열의 선언과 차이점은 없다. 큐의 구조는 배열과 마찬가지로 선형 구조이지만 사용하는 방법이 다를 뿐이다. 배열을 사용할 경우에는 100개의 공간 중에 원하는 곳이라면, 어디에서나 삽입과 삭제 연산을 할 수 있지만 배열을 큐로 사용하면 정해진 입구와 출구를 통해서만 삽입과 삭제 연산을 한다.

큐 출입구의 선언

〈표Ⅲ.7.2〉 큐의 변수 선언

<VC> int rear = -1, front = -1;
<VB> Dim REAR As Integer, FRONT As Integer

〈표Ⅲ.7.3〉 큐의 초기 상태

<VC> 현재 큐의 값()

0	1	2	3	4	5	6	7	…	99	Queue

$\boxed{-1}$ rear (현재 큐의 입구를 나타냄) $\boxed{-1}$ front (현재 큐의 출구를 나타냄)
// 입력 할 때에는 rear을 1씩 증가시키고, 삭제 할 때에는 front를 1씩 증가.

<VB> 현재 큐의 값()

1	2	3	4	5	6	7	8	…	100	QUEUE

$\boxed{0}$ REAR (현재 큐의 입구를 나타냄) $\boxed{0}$ FRONT (현재 큐의 출구를 나타냄)
' 입력 할 때에는 REAR을 1씩 증가시키고, 출력 할 때에는 FRONT를 1씩 증가

이 상태에서 10을 삽입하게 되면 큐의 상태는 〈표Ⅲ.7.4〉과 같이 된다.

〈표Ⅲ.7.4〉 큐에 하나의 자료를 삽입한 상태

<VC> 현재 큐의 값(10)

10									
0	1	2	3	4	5	6	7	…	99

Queue

0 rear, -1 front

<VB> 현재 큐의 값(10)

10									
1	2	3	4	5	6	7	8	…	100

QUEUE

1 REAR, 0 FRONT

연속해서 3, 7, 2를 입력하게 되면 스택은 아래와 같은 상태가 된다.

〈표Ⅲ.7.5〉 큐에 4개의 값을 삽입한 상태

<VC> 현재 큐의 값(10, 3, 7, 2)

10	3	7	2						
0	1	2	3	4	5	6	7	…	99

Queue

3 rear, -1 front

<VB> 현재 큐의 값(10, 3, 7, 2)

10	3	7	2						
1	2	3	4	5	6	7	8	…	100

QUEUE

4 REAR, 0 FRONT

위의 그림에서 알 수 있듯이, Rear값은 큐의 입구를 나타내고 있고, 배열을 진한 회색으로 표현하였다. Front값보다 1칸 뒤는 큐의 출구를 나타내고 배열을 연한 회색으로 표현하였다. 즉, 입구에 자료 2가 있고, 출구에는 자료 10이 있다는 의미이다. 자료를 삭제하게 되면, 10이 제거될 것이고 큐의 출구는 한 칸 이동한다. 〈표Ⅲ.7.5〉에서 삭제 연산을 수행하면 결과는 〈표Ⅲ.7.6〉과 같다.

〈표Ⅲ.7.6〉 큐에서 값을 삭제

<VC> 현재 큐의 값(3, 7, 2)

10	3	7	2							
0	1	2	3	4	5	6	7	…	99	Queue

3 rear, 0 front

<VB> 현재 큐의 값(3, 7, 2)

10	3	7	2							
1	2	3	4	5	6	7	8	…	100	QUEUE

4 REAR, 1 FRONT

위의 결과를 보면 10을 삭제했는데 배열에는 10이 그대로 남아있지만 10은 front보다 앞에 존재하므로 큐에 없다고 생각해도 된다.

위의 예에서 제시된 내용을 큐 연산으로 표현해 보면 〈표Ⅲ.7.7〉과 같다.

〈표Ⅲ.7.7〉 큐 연산

```
<VC>  Push(10);
      Push(3);
      Push(7);
      Push(2);
      printf("%d",Pop());     // 큐의 값을 제거하면서 화면에 출력하는 역할
          // Push(a)는 a를 큐에 삽입하라는 연산
          // Pop()은 큐에서 값을 제거해서 반환하라는 연산
```

```
<VB>  Push 10
      Push 3
      Push 7
      Push 2
      Magbox Pop()     '큐의 값을 제거 하면서 메시지 박스로 출력

      'Push a는 a를 큐에 삽입하라는 연산
      'Pop()은 큐에서 값을 제거해서 반환하라는 연산
```

여기서 화면에 출력되는 값은 스택과는 다르게 10이 된다. FIFO이므로 먼저 입력된 자료가 먼저 나온다.

예제Ⅲ.7.1

아래와 같이 큐에서의 작업이 정의될 때 다음 중 큐 오버플로우(overflow)가 일어나는 경우는? (2002 중등)

큐에서의 작업(operation)
1. create(q, n) : 크기가 n인 빈 큐 q를 만든다.
2. insert(q, item) : 큐 q에 item을 삽입한다.
3. remove(q) : 큐 q에서 값을 하나 삭제한다.

① create(q, 1)
insert(q, 1)
insert(q, 2)

② create(q, 2)
insert(q, 1)
insert(q, 2)

③ create(q, 3)
insert(q, 1)
insert(q, 2)

④ create(q, 2)
insert(q, 1)
remove(q)

⑤ create(q, 2)
remove(q)
insert(q, 2)

<풀이> ①

큐 오버플로우도 스택 오버플로우처럼 큐에 자료가 더 이상 넣을 공간이 없는데 삽입연산을 할 경우 발생한다.

문제Ⅲ.7.1

'이상한' 바이러스는 정상적인 환경에서는 1초마다 현재 수의 2배로 불어난다 그러나 환경이 갑자기 변하면 1초마다 현재의 $\frac{1}{3}$로 줄어들기도 한다.(소숫점 이하는 버린다.) 예를 들어 10마리의 바이러스가 있다면 1초 후에는 20마리(10 ×2 = 20)가 될 수도 있고, 3마리(10 ÷3 = 3.333…)가 될 수도 있다. 현재 6마리의 바이러스가 있는데, 최초 1마리의 '이상한' 바이러스로부터 어떻게 현재 6마리가 되었는지 알아보기 위해 큐를 사용하여 다음과 같은 알고리즘으로 프로그램을 작성하였다.

㉠ 큐에 1을 삽입(insert)한다.
㉡ 큐에서 하나를 제거(remove)한 뒤 그 수에 2를 곱한 수와 $\frac{1}{3}$을 곱한 수를 차례로 큐에 삽입한다.
㉢ ㉡의 과정을 큐에 삽입되는 수가 6일 때까지 반복한다. 큐에 삽입되는 수가 6이면 그 때까지 바이러스의 수가 변화한 과정을 찾아 출력한다.

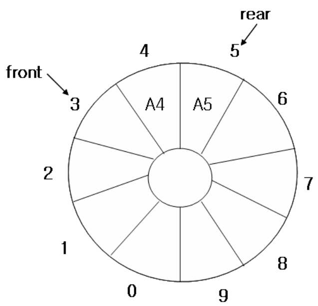
이와 같이 해서 1마리에서 6마리가 되는 과정을 찾아낼 때, 그 결과로 올바른 것은? (2002 고등)

① 1 → 2 → 4 → 8 → 16 → 5 → 10 → 3 → 6
② 1 → 2 → 4 → 8 → 16 → 5 → 10 → 20 → 6
③ 1 → 2 → 4 → 8 → 16 → 32 → 10 → 3 → 6
④ 1 → 2 → 4 → 8 → 16 → 32 → 10 → 20 → 6
⑤ 1 → 2 → 4 → 8 → 16 → 32 → 64 → 20 → 6

<풀이> ③

예제Ⅲ.7.2

크기가 10 이고 front가 3, rear가 5인 다음의 원형 큐에서 새로운 항목이 하나 삽입되었을 경우, front와 rear의 새로운 값은?(2006 교원11번)



(front는 삭제가 일어나는 곳을 말하며, 항상 첫 번째 원소로부터 시계반대방향으로 하나 앞 위치를 가리키게 한다. rear는 삽입이 일어나는 곳을 말하며 큐의 현재 데이터 끝을 가리킨다. 위 그림은 A4, A5의 데이터가 현재 큐에 있음을 보여주며 front와 rear의 현 위치를 보여준다.)

① front = 4, rear = 5 ② front = 3, rear = 6
③ front = 4, rear = 6 ④ front = 2, rear = 4
⑤ front = 2, rear = 6

<풀이> ②
원형 큐에서는 새로운 항목을 하나 삽입하면 rear의 값이 1증가한다. 큐에서는 자료를 추출하고 저장할 때마다 Front와 Rear은 각각 1씩 더해지므로 언젠가는 배열의 마지막 요소에 도달한다. 그러나 큐에서 자료를 추출한 후 head보다 왼쪽에 있는 배열 요소는 비어있으므로 저장 공간의 낭비를 초래한다. 배열의 끝 queue[n-1]과 처음 queue[0]을 연결해서 원형의 배열을 생각할 수 있다. 원형 큐에서는 queue[n-1]까지 큐를 저장하고 다음 자료를 저장할 때에는 다시 queue[0], queue[1], …에 저장하게 된다.

연산자의 구현

〈표Ⅲ.7.8〉 큐 연산자 구현

```
<VC> void Push(int data){
       Queue[++rear] = data;             //큐에 데이터를 삽입
       }


       int Pop(void){
            return Queue[++front] ;           //큐의 값을 제거, 반환
       }
   // 스택과 달리 큐의 연산에서는 rear, front 모두 먼저 1증가시키는 것에 유의
```

```
<VB> Sub Push(data as integer)
       REAR = REAR + 1
       QUEUE(REAR) = data       '큐에 값을 삽입
       End Sub

       Function Pop() As Integer
          FRONT = FRONT + 1
          Pop = QUEUE(FRONT)      '큐에서 값을 제거, 반환
       End Function
```

연 습 문 제

1 다음의 빈칸에 적합한 단어는 무엇인가?

(　　　)은 rear라는 한 쪽 끝에서 삽입이 일어나고 front라는 다른 쪽 끝에서 삭제가 일어나는 선형 자료구조이다.

① 큐　② 트리　③ 스택　④ 그래프　⑤ 연결리스트

2 선형 자료 구조와 비선형 자료 구조로 바르게 짝지어진 것은?

	선형 자료 구조	비선형 자료 구조
①	트리, 연결 리스트	스택, 큐, 그래프
②	스택, 큐, 연결 리스트	트리, 그래프
③	연결 리스트, 그래프	스택, 큐, 트리
④	스택, 큐, 연결 리스트, 그래프	트리
⑤	스택, 큐, 트리, 연결 리스트	그래프

3 양방향에서 입출력이 가능한 선형 자료구조로서 2개의 포인터를 이용하여 양쪽 끝 모두에서 삽입·삭제가 가능한 것은?

① 큐(Queue)　② 트리(Tree)　③ 스택(Stack)
④ 그래프(Graph)　⑤ 데크(Deque)

4 자료구조에 관한 설명 중 옳지 않은 것은?

① 스택은 후입선출구조(LIFO)로 복귀주소(return address) 등에 이용된다.
② 큐는 선입선출구조(FIFO)로 작업 스케줄링 등에 이용된다.
③ 트리는 선형 자료구조이다.
④ 데크(Deque)는 서로 다른 방향에서 입・출력이 가능한 구조이다.
⑤ 그래프는 비선형 자료구조이다.

5 운영체제의 작업 스케줄링 등에 응용될 수 있는 가장 적합한 자료구조는?

① 큐　② 트리　③ 스택　④ 그래프　⑤ 연결리스트

6 배열로 구현된 큐 A에 다음과 같이 삽입(push)과 삭제(pop)가 되풀이 된 후 큐에 남아있는 값들의 합은 얼마인가?

push(A,1) ⇒ pop(A,2) ⇒ push(A,3) ⇒ pop(A) ⇒ push(A,4) ⇒ pop(A) ⇒ pop(A) ⇒ push(A,5)

① 6　② 7　③ 8　④ 9　⑤ 10

풀 이

1 ①

2 ②

스택, 큐, 연결 리스트는 선형 자료 구조이고 트리, 그래프는 비선형 자료 구조이다.

3 ⑤

4 ③

5 ①

스택과 마찬가지로 큐도 자료의 임시저장에 많이 사용되는데, 스택은 출입구가 하나라서 가장 나중에 들어간 자료가 가장 먼저 나오는 반면에 큐는 입구와 출구가 하나씩 존재하기 때문에 가장 먼저 들어간 자료가 가장 먼저 나온다. 큐가 가장 많이 사용되는 부분은 키보드 버퍼이다. 가장 먼저 입력된 키부터 우선 처리한다. 운영체제의 작업 스케줄링도 먼저 처리해야 하는 작업부터 우선 처리되어야 하므로 큐를 이용한다.

6 ④

삽입과 삭제 후의 큐의 상태는 다음과 같다. 마지막으로 큐에 남아있는 값들의 합은 9가 된다.

1	2	3		
		↓		
	2	3		
		↓		
	2	3	4	
		↓		
		3	4	
		↓		
			4	5