

2012-계리직-컴퓨터일반-A형-해설

대방고시 전산직/계리직, 하이클래스 군무원 곽후근(gobarian@gmail.com)
해설에 대한 모든 권리는 곽후근(대방고시, 하이클래스)에 있습니다.

1. 컴퓨터 용어에 대한 설명으로 옳지 않은 것은?

- ① MIPS는 1초당 백만개 명령어를 처리한다는 뜻으로 컴퓨터의 연산 속도를 나타내는 단위이다.
- ② SRAM은 전원이 꺼져도 저장된 자료를 계속 보존할 수 있는 기억장치이다.
- ③ KB, MB, GB, TB 등은 기억 용량을 나타내는 단위로서 이중 TB가 가장 큰 단위이다.
- ④ SSI, MSI, LSI, VLSI 등은 칩에 포함되는 게이트의 집적도에 따라 구분된 용어이다.

정답 체크 :

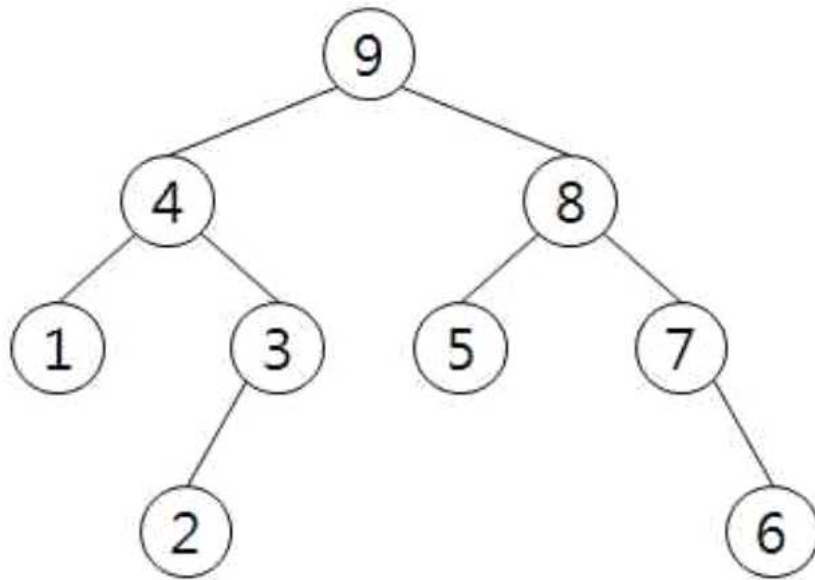
(2) MIPS : 초당 처리하는 명령어의 개수로 단위는 1초에 100만 개의 명령어 수행을 나타내는 MIPS(Million Instruction Per Second)으로 나타낸다. 처리속도가 18.5MIPS라면 1초 동안에 1,850만 개의 명령을 실행할 수 있다. (초당 처리하는 명령어의 개수가 많아 100만 개 단위로 연산 속도를 나타낸다. 초당 부동 소수점 수(명령어 중에 처리 시간이 가장 오래 걸림)의 처리 개수는 나타내는 FLOPS를 같이 사용한다.)

오답 체크 :

- (1) SRAM은 정적 RAM으로 캐시에 사용되고, 전원이 꺼지면 저장된 자료가 없어지는 휘발성이다. 참고로, 현재 개발 중인 비휘발성 RAM에는 FRAM, MRAM, PRAM 등이 있다.
 - (3) 저장용량을 순서대로 나열하면 다음과 같다 : bit < Byte < KB < MB < GB < TB < PB
 - (4) 집적도를 순서대로 나열하면 다음과 같다 : SSI < MSI < LSI < VLSI < ULSI
- Tip! : 저장용량의 경우 더 작은 단위와 더 큰 단위로 시험에 자주 출제되므로 기억해두는 것이 좋다.

10^n	접두어	기호	배수	십진수
10^{24}	요타 (yotta)	Y	자	1 000 000 000 000 000 000 000 000
10^{21}	제타 (zetta)	Z	십해	1 000 000 000 000 000 000 000
10^{18}	엑사 (exa)	E	백경	1 000 000 000 000 000 000
10^{15}	페타 (peta)	P	천조	1 000 000 000 000 000
10^{12}	테라 (tera)	T	조	1 000 000 000 000
10^9	기가 (giga)	G	십억	1 000 000 000
10^6	메가 (mega)	M	백만	1 000 000
10^3	킬로 (kilo)	k	천	1 000
10^2	헥토 (hecto)	h	백	100
10^1	데카 (deca)	da	십	10
10^0			일	1
10^{-1}	데시 (deci)	d	십분의 일	0.1
10^{-2}	센티 (centi)	c	백분의 일	0.01
10^{-3}	밀리 (milli)	m	천분의 일	0.001
10^{-6}	마이크로 (micro)	μ	백만분의 일	0.000 001
10^{-9}	나노 (nano)	n	십억분의 일	0.000 000 001
10^{-12}	피코 (pico)	p	일조분의 일	0.000 000 000 001
10^{-15}	펨토 (femto)	f	천조분의 일	0.000 000 000 000 001
10^{-18}	아토 (atto)	a	백경분의 일	0.000 000 000 000 000 001
10^{-21}	젠티 (zepto)	z	십해분의 일	0.000 000 000 000 000 000 001
10^{-24}	욕토 (yocto)	y	일자분의 일	0.000 000 000 000 000 000 000 001

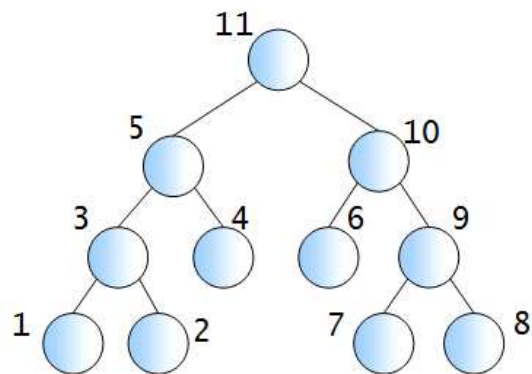
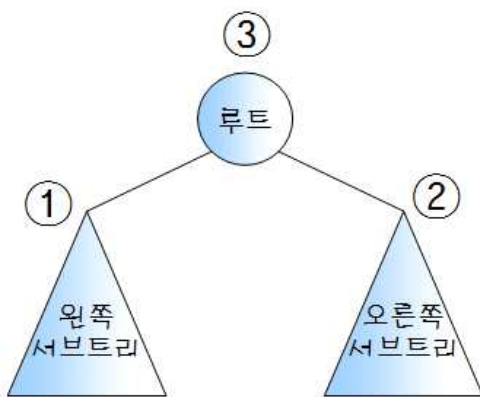
2. 이진트리의 순회(traversal) 경로를 나타낸 그림이다. 이와 같은 이진트리 순회방식은 무엇인가? 단, 노드의 숫자는 순회순서를 의미한다.



- ① 병렬 순회(parallel traversal)
- ② 전위 순회(pre-order traversal)
- ③ 중위 순회(in-order traversal)
- ④ 후위 순회(post-order traversal)

정답 체크 :

(4) 후위 순회 : 해당 순회는 후위 순회를 나타내고, 후위 순회를 일반화해서 그림으로 나타내면 다음과 같다.



오답 체크 :

(1) 병렬 순회 : GPU에서 사용하는 병렬 순회를 나타낸다. 병렬 순회는 여러 개의 프로세서가 동시에 트리를 병렬로 순회함을 의미한다. 병렬 순회를 그림으로 나타내면 다음과 같다.

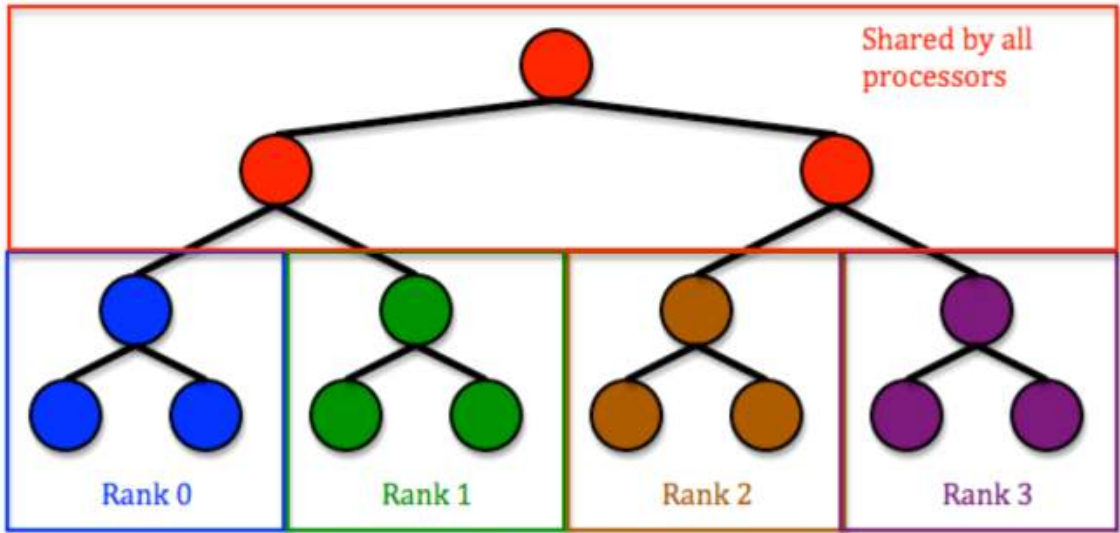
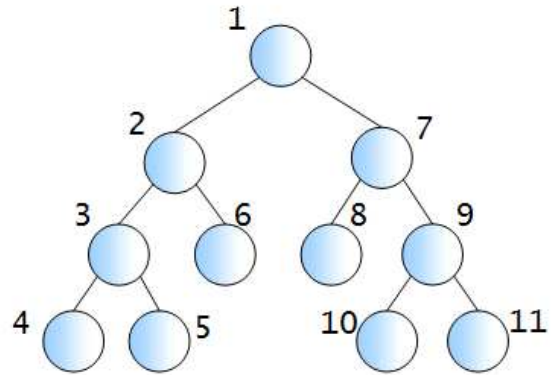
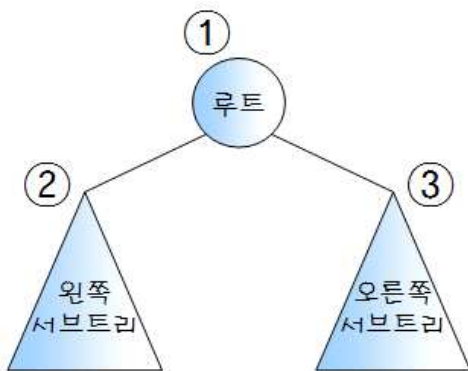
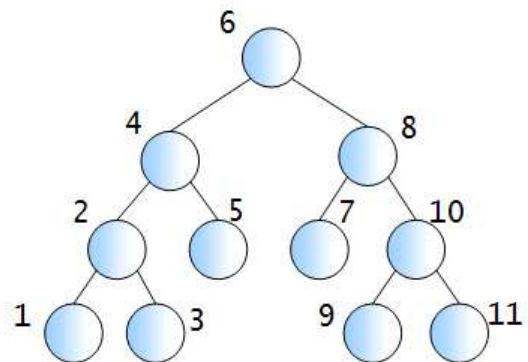
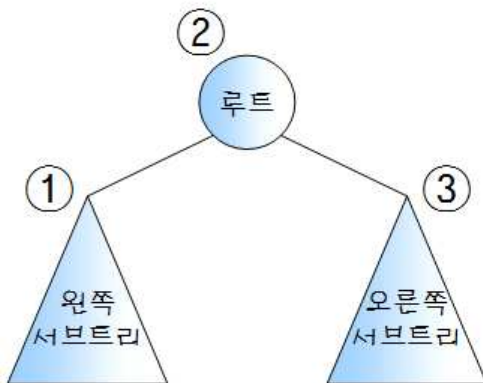


Figure 2: A tree of height 4 split among 4 processors

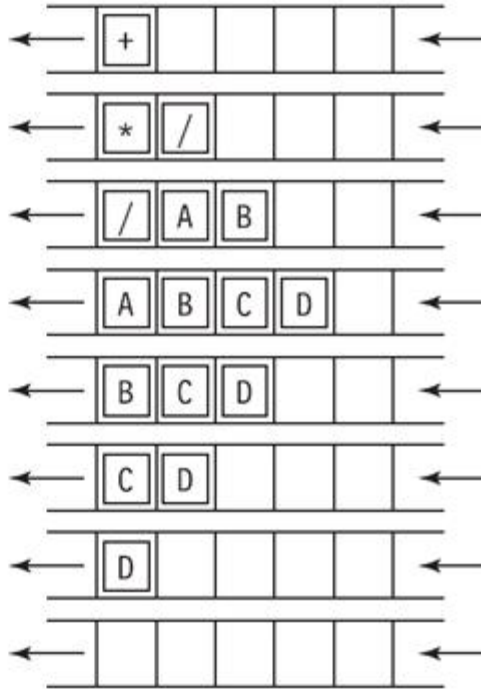
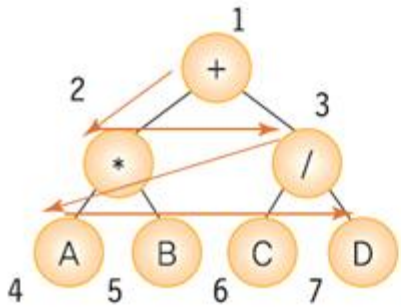
(2) 전위 순회 : 주어진 트리를 전위 순회로 방문하면 (9, 4, 1, 3, 2, 8, 5, 7, 6)이 되고, 전위 순회를 일반화해서 그림으로 나타내면 다음과 같다.



(3) 중위 순회 : 주어진 트리를 중위 순회로 방문하면 (1, 4, 2, 3, 9, 5, 8, 7, 6)이 되고, 중위 순회를 일반화해서 그림으로 나타내면 다음과 같다.



Tip! : 이외에도 레벨 순회(level order traversal)가 존재한다. 레벨 순회는 각 노드를 레벨 순으로 검사하는 순회 방법으로 전위/중위/후위 순회가 스택을 사용한다면, 레벨 순회를 큐를 사용해서 순회를 한다. 레벨 순회를 그림으로 나타내면 다음과 같다.



3. 엑셀에서는 서로 다른 시트 사이에 셀 참조가 가능하다. 아래 그림에서 Sheet2의 시금치 가격을 VLOOKUP 함수를 사용하여 Sheet1에서 가져오려고 한다. 이를 위해 Sheet2의 B3 셀에 입력할 수식으로 알맞은 것은?

Sheet1

	A	B	C	D
1	상품명	산지	생산자	가격
2	오이	청주	김철수	500
3	배추	울산	황인용	2000
4	무우	김제	김영운	1500
5	시금치	평창	나윤로	1000
6	상추	대전	김윤철	700

Sheet2

	A	B
1	상품명	가격
2	무우	
3	시금치	
4		
5		
6		

- ① =VLOOKUP(시금치,Sheet1!A2:D6,4,0)
- ② =VLOOKUP(시금치,A2:A6,5,0)
- ③ =VLOOKUP(A3,Sheet1!A2:D6,4,0)
- ④ =VLOOKUP(A3,Sheet1!A2:A6,5,0)

정답 체크 :

(3)

VLOOKUP(lookup_value, table_array, col_index_num, [range_lookup])의 사용법을 설명하면 다음과 같다.

lookup_value : 둘째 인수인 표(Sheet2)의 1열에서 찾을 값이므로 A3가 된다. 아니면 직접

“시금치”로 입력해야 한다(텍스트임을 알리기 위해 큰따옴표로 묶어야 한다).

table_array : 찾는 값과 함수 결과값이 모두 있는 참조표의 범위이므로 Sheet1!A2:D6이다. Sheet1!이 붙은 이유는 참조표가 Sheet2와 다른 Sheet1에 있기 때문이고, 범위는 A2에서 D6까지가 된다.

col_index_num : 찾는 값을 표(Sheet2)의 첫 열에서 찾았을 때 실제 함수의 결과값(Sheet1)이 있는 열의 번호이므로 4열이 된다.

range_lookup : 범위를 이용한 검색을 하지 않을 것(정확히 일치하는 값을 찾을 때)이므로 FALSE(0)으로 지정한다.

오답 체크 :

- (1) 시금치를 “시금치”로 수정해야 한다.
- (2) 시금치를 “시금치”로 수정하고, A2:A6를 Sheet1!A2:A6로 수정하고, 5를 4로 수정해야 한다.
- (4) 5를 4로 수정해야 한다.

Tip! : VLOOKUP 함수와 사용 예를 정리하면 다음과 같다.

VLOOKUP 찾는 값과 관련된 결과값을 표 범위의 특정 열(결과값의 열 번호)에서 가져온다.

VLOOKUP(lookup_value, table_array, col_index_num, [range_lookup])

찾는 값: 둘째 인수인 표의 1열에서 찾을 값

표 범위: 찾는 값과 함수 결과값이 모두 있는 참조표의 범위(범위 검색 여부가 TRUE이면 표의 첫 열은 오름차순으로 정렬되어야 함)

결과값의 열 번호: 찾는 값을 표의 첫 열에서 찾았을 때 실제 함수의 결과값이 있는 열의 번호

범위 검색: 범위를 이용한 검색 여부를 TRUE/FALSE로 지정(TRUE는 생략 가능)

그림 5-2 품명을 직접 입력하기와 함수를 이용해 입력하기의 비교

	A	B	C	D	E	F	G	H	I	J
1	판매일	제품코드	품명	단가	개수	판매액		품목별 단가표		
2	3월 1일	OP-A4	A4 복사지	19,000	30	570,000				
3	3월 1일	OP-A3	3		20	-				
4	3월 1일	OP-B5			10	-				
5	3월 2일	ON-SB5			7	-				
6	3월 2일	OP-A4			32	-				
7	3월 2일	ON-B5			11	-				
8	3월 3일	OP-A4			27	-				

코드	동명	단가
ON-B5	B5 노트	10,000
ON-SB5	스프링 노트	20,000
OP-A3	A3 복사지	28,000
OP-A4	A4 복사지	19,000
OP-B5	B5 복사지	17,000

	품명 입력하기	단가 입력하기
직접 입력 과정	<ol style="list-style-type: none"> ① [B2] 셀의 OP-A4를 품목별 단가표(H4:J8)의 1열에서 찾은 후 ② 품목별 단가표(H4:J8)의 2열에서 해당 품명이 A4 복사지임을 확인하고 ③ [C2] 셀에 A4 복사지를 입력한다. 	<ol style="list-style-type: none"> ④ [B2] 셀의 OP-A4를 품목별 단가표(H4:J8)의 1열에서 찾은 후 ⑤ 품목별 단가표(H4:J8)의 3열에서 해당 단가가 19,000임을 확인하고 ⑥ [D2] 셀에 19,000을 입력한다.
함수 이용	= VLOOKUP(B2, \$H\$4:\$J\$8, 2, FALSE)	= VLOOKUP(B2, \$H\$4:\$J\$8, 3, FALSE)

4. <보기>는 모듈화를 중심으로 한 소프트웨어 설계방법에 대한 설명이다. 빈칸의 내용을 올바르게 나열한 것은?

<보기>

- 결합도(coupling)와 응집도(cohesion)는 모듈의 (㉠)을 판단하는 기준이다.
- 결합도란 모듈 (㉡)의 관련성을 의미하며, 응집도란 모듈 (㉢)의 관련성을 의미한다.
- 좋은 설계를 위해서는 결합도는 (㉣), 응집도는 (㉤) 방향으로 설계해야 한다.

	㉠	㉡	㉢	㉣	㉤
① 독립성	사이	내부	작게	큰	
② 독립성	내부	사이	크게	작은	
③ 추상성	사이	내부	작게	큰	
④ 추상성	내부	사이	크게	작은	

정답 체크 :

(1)

결합도와 응집도는 모듈의 독립성을 판단하는 두 가지 지표이다. 결합도는 모듈과 모듈간의 상호 의존 정도, 응집도는 모듈 내부의 기능적인 집중 정도라고 할 수 있다.

결합도(coupling)는 모듈과 모듈 사이의 관계에서 관련 정도를 의미한다. 하나의 모듈 안에서 구성 요소들 간에 푼푼 뭉쳐 있는 정도로 평가한다. 데이터(자료) 결합은 모듈들이 매개 변수를 통해 데이터만 주고받음으로써 서로 간섭을 최소화하는 관계한다. 모듈 간의 독립성 보장한다. 내용 결합은 모듈 간에 인터페이스를 사용하지 않고 직접 왔다 갔다 하는 경우의 관계이다. 상대 모듈의 데이터를 직접 변경할 수 있어 서로 간섭을 가장 많이 하는 관계이다.

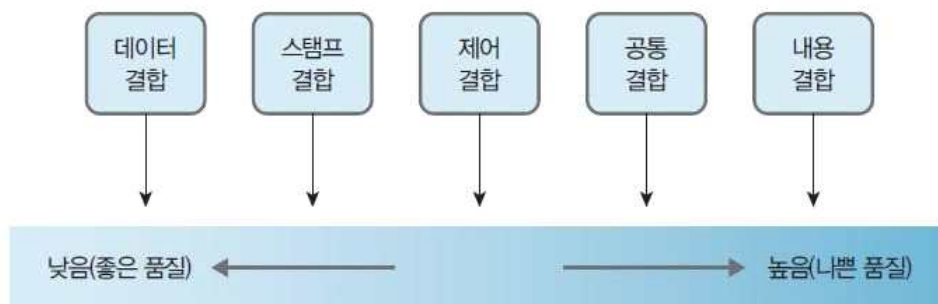


그림 6-11 모듈 간의 결합도

응집도(Cohesion)란 모듈 내부에 존재하는 구성 요소들 사이의 밀접한 정도이다. 하나의 모듈 안에서 구성 요소들 간에 푼푼 뭉쳐 있는 정도로 평가한다. 기능적 응집은 함수적 응집으로 응집도가 가장 높은 경우이며 단일 기능의 요소로 하나의 모듈을 구성한다. 우연적 응집은 구성 요소들이 말 그대로 우연히 모여 구성된 것이다. 특별한 이유 없이, 크기가 커 몇 개의 모듈로 나누는 과정에서 우연히 같이 묶인 것이다.

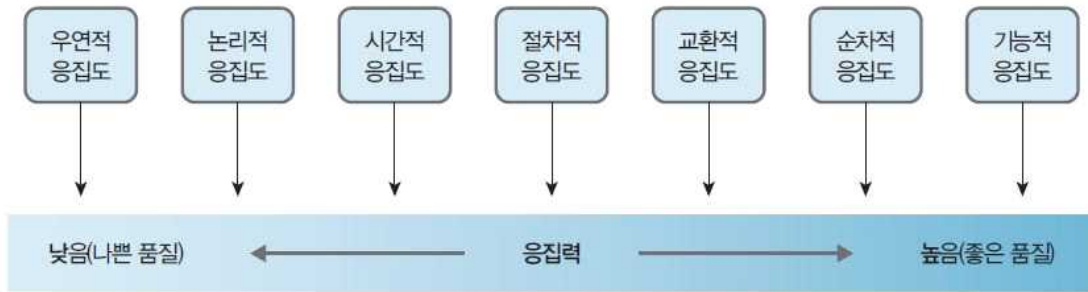


그림 6-2 모듈 내 구성 요소 간의 응집도

오답 체크 :

(2) 독립성은 맞으나, 나머지는 반대로 되어 있다.

(3), (4) 추상성 : 모듈화가 아닌 캡슐화에 사용되는 용어로, 객체의 속성, 오퍼레이션 등의 세부사항(구현)은 차후에 생각하는 것을 의미한다.

5. 다음 중 데이터 값의 대소를 비교하여 정렬하는 문제에 대한 가장 빠른 알고리즘의 시간복잡도는? 단, n 은 정렬 대상의 입력 데이터 수이다.

- ① $O(n)$
- ② $O(\log_2 n)$
- ③ $O(n \log_2 n)$
- ④ $O(n^2)$

정답 체크 :

(3) 정렬 알고리즘의 시간 복잡도를 정리하면 다음과 같다. 시험에 자주 출제되므로 무조건 숙지하여야 한다. 시간 복잡도는 직접 구현하지 않고서도 수행 시간을 분석하는 것이다. 예를 들어, 입력의 개수 n 에 대해 연산(비교, 이동 등)의 횟수를 함수로 나타낸다. 빅오(O)는 연산의 횟수를 대략적(점근적/근사적)으로 표기한 것으로 함수의 상한을 표시한다. 가장 빠른 알고리즘의 시간복잡도는 삽입 정렬의 최선에서 $O(n)$ 이지만 이는 이미 정렬된 데이터를 기준으로 하기 때문에 보기에서 제외한다. 즉, 주어진 문제의 조건(데이터 값의 대소를 비교하여 정렬)은 평균, 최악이라고 가정하면 가장 빠른 알고리즘의 시간 복잡도는 $O(n \log_2 n)$ 이 된다.

알고리즘	최선	평균	최악
삽입 정렬	$O(n)$	$O(n^2)$	$O(n^2)$
선택 정렬	$O(n^2)$	$O(n^2)$	$O(n^2)$
버블 정렬	$O(n^2)$	$O(n^2)$	$O(n^2)$
셸 정렬	$O(n)$	$O(n^{1.5})$	$O(n^2)$
퀵 정렬	$O(n \log_2 n)$	$O(n \log_2 n)$	$O(n^2)$
힙 정렬	$O(n \log_2 n)$	$O(n \log_2 n)$	$O(n \log_2 n)$
합병 정렬	$O(n \log_2 n)$	$O(n \log_2 n)$	$O(n \log_2 n)$
기수 정렬	$O(dn)$	$O(dn)$	$O(dn)$

오답 체크 :

- (1) 정렬 알고리즘 시간 복잡도(삽입 정렬의 최선)는 맞으나 문제의 조건을 의미하지 않는다.
- (2) 정렬 알고리즘 시간복잡도가 아니다.
- (4) 정렬 알고리즘 시간복잡도는 맞으나 (3)번에 비해 오래 걸린다.

6. 여덟 개의 페이지(0 ~ 7페이지)로 구성된 프로세스에 네 개의 페이지 프레임이 할당되어 있고, 이 프로세스의 페이지 참조 순서는 <보기>와 같다. 이 경우 LRU 페이지 교체 알고리즘을 적용할 때 페이지 적중률(hit ratio)은 얼마인가? 단, <보기>의 숫자는 참조하는 페이지번호를 나타내고, 최초의 페이지 프레임은 모두 비어있다고 가정한다.

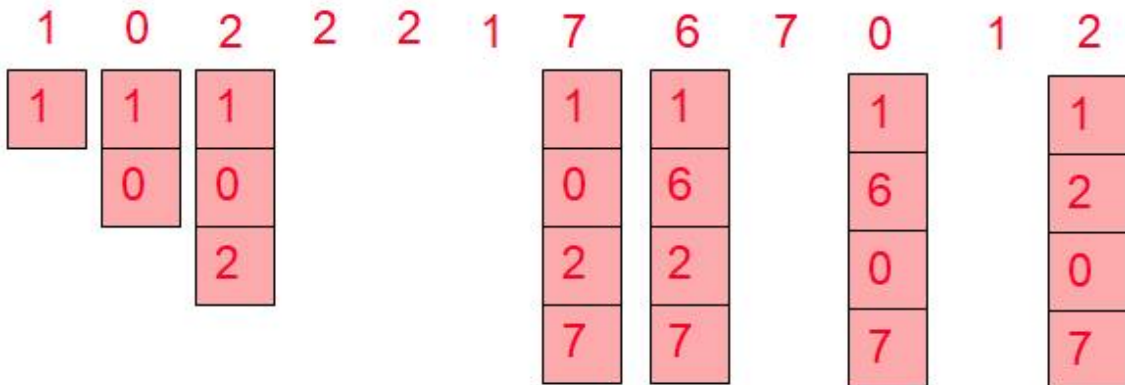
<보기>
1, 0, 2, 2, 2, 1, 7, 6, 7, 0, 1, 2

- ① 5/12
- ② 6/12
- ③ 7/12
- ④ 8/12

정답 체크 :

(1)

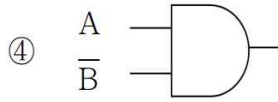
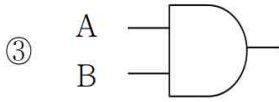
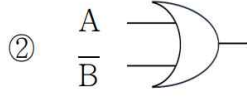
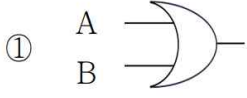
LRU는 프로세스가 가장 최근의 페이지에 액세스했다는 것은 멀지 않아 다시 액세스할 가능성이 있다는 의미이다. 과거 오랫동안 사용하지 않은 페이지로 대체하는 효과로 생각할 수 있다. 페이지를 대체할 때 오랫동안 사용하지 않은 페이지를 선택하므로 시간적으로 거꾸로 찾는 최적 페이지 대체 알고리즘이라고 할 수 있다. 주어진 조건으로 LRU를 적용하면 다음과 같다. 8번째 6이 들어왔을 때 페이지 프레임 내에 6이 없기 때문에 교체를 해야 한다. 이때, 7이 가장 최근에 사용되었고, 그 다음에 1이, 그 다음에 2가 사용되었기 때문에 0을 교체한다. 페이지가 사용되었다는 의미는 참조열이 프레임 내의 빈 공간에 새로 들어왔거나 참조열과 동일한 프레임이 존재하는 것을 의미한다. 이러한 방식으로 참조열을 처리하면 총 5회의 페이지 적중이 발생한다. 총 참조 횟수가 12번이고, 적중 횟수가 5번이므로 적중률은 5/12가 된다.



7. <보기>의 논리 연산식을 간략화한 논리회로는?

<보기>

$$(A+B)(A+B')(A'+B)$$



정답 체크 :

(3)

연산식을 두 개로 나눠서 간략화하면 다음과 같다.

$$(A+B)(A+B') = A + AB' + AB + BB' = A(1 + B' + B) + 0 = A1 = A$$

$$A(A'+B) = AA' + AB = 0 + AB = AB$$

8. <보기>의 설명에 해당하는 네트워크 장비는?

<보기>

- OSI 계층 모델의 네트워크 계층에서 동작하는 장비이다.
- 송신측과 수신측 간의 가장 빠르고 신뢰성 있는 경로를 설정·관리하며, 데이터를 전달하는 역할을 한다.
- 주로 같은 프로토콜을 사용하는 네트워크간의 최적경로 설정을 위해 패킷이 지나가야 할 정보를 테이블에 저장하여 지정된 경로를 통해 전송한다.

- ① 게이트웨이(gateway)
- ② 브리지(bridge)
- ③ 리피터(repeater)
- ④ 라우터(router)

정답 체크 :

(4) 라우터 : 3계층(네트워크 계층)에서 동작하는 장비이다. 라우터는 LAN, MAN, WAN과 같은 네트워크를 서로 연결해 주는 장비로, 네트워크에 흐르는 패킷의 논리 주소(IP 주소)에 따라 패킷을 라우팅 해준다.

오답 체크 :

(1) 게이트웨이 : 이종의 프로토콜을 서로 연결해주어야 하기 때문에 OSI 계층 모델의 전 계층에서 동작하는 장비이다. 게이트웨이는 다른 네트워크로 들어가는 입구 역할을 하거나 나가는 출구 역할을 하는 장비이다.

(2) 브리지 : 2계층(데이터 링크 계층)에서 동작하는 장비이다. 브리지는 동일한 기관의 두 개 이상의 LAN의 분할된 세그먼트를 서로 연결하여 하나의 네트워크로 만드는 장비로, 네트워크에 흐르는 프레임의 물리주소를 필터링 한다.

(3) 리피터 : 1계층(물리 계층)에서 동작하는 장비이다. 리피터는 네트워크의 전송거리를 연장하기 위하여 사용하는 장비로, 장거리 전송으로 인해 약해진 신호를 재생하여 전송해 준다.

Tip! : 4계층(전송 계층)에서 동작하는 장비에서 부하분산기(L4 스위치), 방화벽 등이 있고, 7계층에서 동작하는 장비에는 IDS, IPS, DPI 등이 있다.

9. 다음 C 프로그램의 실행 결과로 옳은 것은?

```
void main()
{
    int a[4]={10, 20, 30};
    int *p = a;
    p++;
    *p++ = 100;
    **++p = 200;
    printf("a[0]=%d a[1]=%d a[2]=%d\n", a[0], a[1], a[2]);
}
```

- ① a[0]=10 a[1]=20 a[2]=30
- ② a[0]=10 a[1]=20 a[2]=200
- ③ a[0]=10 a[1]=100 a[2]=30
- ④ a[0]=10 a[1]=100 a[2]=200

정답 체크 :

(3)

int *p = a; // p와 a는 운명을 같이 한다. p는 a[0]을 가리킨다.

p++; // p를 하나 증가시키므로 p는 a[1]을 가리킨다.

*p++ = 100; // a[1]에 100을 대입하고 p를 하나 증가시켜 a[2]를 가리키게 한다.

**++p = 200; // p를 하나 증가시켜 a[3]로 만들고 여기에 200을 대입한다.

이들을 정리하면 배열의 각 요소는 다음과 같은 값을 가지게 된다.

a[0] = 10

a[1] = 100

a[2] = 30

a[3] = 200

10. 인터럽트 처리를 위한 <보기>의 작업이 올바른 나열된 것은?

```
<보기>
ㄱ. 인터럽트 서비스 루틴을 수행한다.
ㄴ. 보관한 프로그램 상태를 복구한다.
ㄷ. 현재 수행 중인 명령을 완료하고 상태를 저장한다.
ㄹ. 인터럽트 발생 원인을 찾는다.
```

- ① ㄷ → ㄹ → ㄱ → ㄴ
- ② ㄷ → ㄹ → ㄴ → ㄱ
- ③ ㄹ → ㄷ → ㄱ → ㄴ
- ④ ㄹ → ㄷ → ㄴ → ㄱ

정답 체크 :

(1)

ㄷ. 현재 수행 중인 명령을 완료하고 상태를 저장한다. : 현재의 PC와 PSW(Program Status Word : C(캐리)S(부호)Z(제로)V(오버플로우) 등)를 저장한다.

르. 인터럽트 발생 원인을 찾는다. : 동기 인터럽트(프로그램 오류 등)인지 비동기 인터럽트(입출력에서 발생 등)인지 확인한다.

거. 인터럽트 서비스 루틴을 수행한다. : 해당 인터럽트를 처리할 수 있는 ISR(Interrupt Service Routine)을 수행한다.

나. 보관한 프로그램 상태를 복구한다. : ISR을 마치면 원래 프로그램으로 돌아오기 위해 이전에 저장한 PC와 PSW를 복구한다.

11. <표>의 CPM(Critical Path Method) 소작업 리스트에서 작업 C의 가장 빠른 착수일(earliest start time), 가장 늦은 착수일(latest start time), 여유 기간(slack time)을 순서대로 나열한 것은?

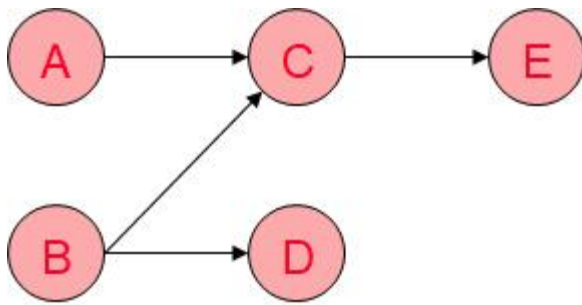
<표> CPM 소작업 리스트

소작업	선행 작업	소요 기간(일)
A	없음	15
B	없음	10
C	A, B	10
D	B	25
E	C	15

- ① 15일, 15일, 0일
- ② 10일, 15일, 5일
- ③ 10일, 25일, 5일
- ④ 15일, 25일, 0일

정답 체크 :

주어진 조건으로 CPM 네트워크를 그리면 다음과 같다.



또한 주어진 조건으로 가장 빠른 착수일과 가장 느린 착수일을 계산하면 다음과 같다.

가장 빠른 착수일				
A	B	C	D	E
0	0	15	10	25
15	10	10	25	15

가장 늦은 착수일				
A	B	C	D	E
0	5	15	15	25
15	10	10	25	15

위의 표를 기준으로 작업 C의 가장 빠른 착수일은 15일이 되고, 가장 늦은 착수일은 15이 된다. 여유 기간은 (가장 늦은 착수일 - 가장 빠른 착수일)이므로 0이 된다.

12. <보기>는 스택을 이용한 0-주소 명령어 프로그램이다. 이 프로그램이 수행하는 계산으로 옳은 것은?

```

<보기>
PUSH C
PUSH A
PUSH B
ADD
MUL
POP Z

```

- ① $Z = C + A * B$
- ② $Z = (A + B) * C$
- ③ $Z = B + C * A$
- ④ $Z = (C + B) * A$

정답 체크 :

(2)

문제의 주어진 조건을 스택으로 나타내면 다음과 같다. C, A, B를 스택에 넣는다(push). +는 최상위 데이터 2개를 꺼내(pop) 계산하고(A+B) 이를 다시 스택에 넣는다(push). *를 만나면 최상 데이터 2개를 꺼내(pop) 계산하고((A+B)*C) 이를 다시 스택에 넣는다(push). 스택에서 하나의 데이터((A+B)*C)를 꺼낸다(pop).

		B		
	A	A	A+B	
C	C	C	C	(A+B)*C

13. 트랜잭션의 특성과 이에 대한 설명으로 옳지 않은 것은?

- ① 원자성(atomicity) : 트랜잭션은 완전히 수행되거나 전혀 수행되지 않아야 한다.
- ② 일관성(consistency) : 트랜잭션을 완전히 실행하면 데이터베이스를 하나의 일관된 상태에서 다른 일관된 상태로 바꿔야 한다.
- ③ 고립성(isolation) : 하나의 트랜잭션의 실행은 동시에 실행 중인 다른 트랜잭션의 간섭을 받아서는 안 된다.
- ④ 종속성(dependency) : 완료한 트랜잭션에 의해 데이터베이스에 가해진 변경은 어떠한 고장에도 손실되지 않아야 한다.

정답 체크 :

(4) 해당 설명은 지속성(Durability)에 해당되고, 트랜잭션의 특성에 종속성은 존재하지 않는다. 지속성이란 트랜잭션이 성공적으로 완료된 후 데이터베이스에 반영한 수행 결과는 영구적이어야 함을 의미한다.

오답 체크 :

- (1) 원자성(Atomicity) : 트랜잭션의 연산들이 모두 정상적으로 실행되거나 하나도 실행되지 않아야 하는 all-or-nothing 방식을 의미한다.
- (2) 일관성(Consistency) : 트랜잭션이 성공적으로 수행된 후에도 데이터베이스가 일관성 있는 상태를 유지해야 함을 의미한다.
- (3) 격리성 혹은 고립성(Isolation) : 수행 중인 트랜잭션이 완료될 때까지 다른 트랜잭션들이 중간 연산 결과에 접근할 수 없음을 의미한다.

14. <보기>의 다양한 진법으로 표현한 숫자들을 큰 숫자부터 나열한 것은?

<보기> ㄱ. $F9_{16}$ ㄴ. 256_{10} ㄷ. 11111111_2 ㄹ. 370_8
--

- ① ㄱ, ㄴ, ㄷ, ㄹ
- ② ㄴ, ㄷ, ㄱ, ㄹ
- ③ ㄷ, ㄹ, ㄱ, ㄴ
- ④ ㄹ, ㄱ, ㄴ, ㄷ

정답 체크 :

(2)

모두 하나의 기준(16진수)으로 바꾼다. 주어진 조건에 따라 가장 빠르게 통일할 수 있는 진

수로 바꾼다.

(ㄱ) $F9_{(16)}$

(ㄴ) $256_{(10)} = 100000000_{(2)} = 100_{(16)}$

(ㄷ) $11111111_{(2)} = FF_{(16)}$

(ㄹ) $370_{(8)} = 011111000_{(2)} = F8_{(16)}$

15. 공개키(public key) 암호화 방식에 대한 설명으로 옳지 않은 것은?

- ① 공개키와 개인키로 이루어진다.
- ② 대표적 활용 예로는 전자서명이 있다.
- ③ 송수신자는 서로 다른 키를 사용한다.
- ④ 개인키는 메시지를 전송할 때 사용한다.

정답 체크 :

(4) 공개키는 메시지를 전송할 때(암호화할 때) 사용하고, 개인키는 메시지를 수신할 때(복호화할 때) 사용한다. 개인키는 메시지 전송할 때 사용할 수는 있지만 이는 암호화가 아닌 전자서명을 수행할 때이다.

오답 체크 :

- (1) 공개키와 개인키를 서로 다른 키로서 비대칭키라고도 불린다.
- (2) 대표적 활용 예로는 암호화와 전자서명 등이 있다.
- (3) 송수신자를 서로 다른 키(공개키, 개인키)를 사용한다.

16. 주기억장치와 캐시 기억장치만으로 구성된 시스템에서 <보기>와 같이 기억장치 접근시간이 주어질 때 캐시 적중률 (hit ratio)은?

<보기>

- 평균 기억장치 접근시간 : $T_a = 1.9ms$
- 주기억장치 접근시간 : $T_m = 10ms$
- 캐시 기억장치 접근시간 : $T_c = 1ms$

- ① 80%
- ② 85%
- ③ 90%
- ④ 95%

정답 체크 :

(3)

주기억장치와 캐시기억장치에서 데이터를 인출하는데 소요되는 평균기억장치 접근시간 (Taverage)은 다음과 같다. $T_{average} = H_{hit_ratio} \times T_{cache} + (1 - H_{hit_ratio}) \times T_{main}$ 여기서, H_{hit_ratio} 는 적중률이고, T_{cache} 는 캐시기억장치 접근시간, T_{main} 은 주기억장치 접근시간이다. 주의할 것은 T_{main} 은 주기억장치로 바로 접근하는 시간이 아니고 캐시로 가서 실패한 후에 주기억장치로 가는 시간을 의미한다. 이를 기반으로 문제를 풀면 다음과 같다. 적중률을 x 라고 하면 계산에 의해 90%가 됨을 알 수 있다.

$x1 + (1-x)10 = 1.9$

$10 - 9x = 1.9$

$8.1 = 9x$

x = 0.9

17. <보기>에서 설명하는 객체지향 개념은?

<보기>

출입문 창문 상자

open

- 그림에서 'open'이라는 오퍼레이션(operation)은 객체마다 다르게 기능한다.
- Java 언어에서 오버로딩(overloading), 오버라이딩(overriding)으로 구현되는 개념이다.

- ① 캡슐화(encapsulation)
- ② 인스턴스(instance)
- ③ 다형성(polymorphism)
- ④ 상속(inheritance)

정답 체크 :

(3) 다형성 : overriding(상속 관계에서 성립하고 동일 원형을 가진다.)과 overloading(동일 클래스 혹은 상속 관계에서 성립하고 함수 이름만 같고 반환형과 매개 변수는 틀리다.)를 의미한다.

오답 체크 :

(1) 캡슐화는 다음과 같은 의미를 포함한다.

캡슐화 : 속성과 관련된 오퍼레이션을 클래스 안에 묶어서 하나로 취급하는 것이다.

추상화 : 객체의 속성, 오퍼레이션 등의 세부사항(구현)은 차후에 생각한다.

정보은닉(information hiding) : 캡슐 속에 있는 항목에 대한 정보를 외부에 감추는 것(public, private, protected)이다.

(2) 인스턴스 : 클래스로부터 메모리가 할당된 객체를 의미한다.

(4) 상속 : 하나의 클래스(슈퍼 클래스)를 다른 클래스(서브 클래스)가 상속해서 사용하는 것을 의미한다. 서브 클래스는 상속된 슈퍼 클래스의 멤버 변수와 멤버 함수를 사용할 수 있고, 자신만의 멤버 변수와 멤버 함수를 만들어 사용할 수 있다.

18. <보기>의 연산을 2의 보수를 이용한 연산으로 변환한 것은?

<보기>

$$6_{10} - 13_{10}$$

- ① $00000110_2 + 11110011_2$
- ② $00000110_2 - 11110011_2$
- ③ $11111010_2 + 11110011_2$
- ④ $11111010_2 - 11110011_2$

정답 체크 :

(1)

$$6 - 13 = 6 + (-13)$$

$$6 = 00000110_{(2)}$$

$$13 = 00001101_{(2)}$$

$$13\text{의 } 2\text{의 보수}(-13) : 11110011_{(2)}$$

19. <보기>는 Windows XP의 실행창(시작 ⇒ 실행)에 입력할 수 있는 명령어들을 나열한 것이다. 명령어별로 수행할 수 있는 기능을 순서대로 나열한 것은?

<보기>

dxdiag - msconfig - regedit - mstsc

- ① 컴퓨터사양 확인 - 시작프로그램 편집 - 레지스트리 편집 - 원격데스크탑 실행
- ② 원격데스크탑 실행 - 작업관리자 편집 - 서비스 편집 - 시스템 셋다운 설정
- ③ 컴퓨터사양 확인 - 작업관리자 편집 - 레지스트리 편집 - 원격데스크탑 실행
- ④ 원격데스크탑 실행 - 시작프로그램 편집 - 서비스 편집 - 시스템 셋다운 설정

정답 체크 :

(1)

dxdiag : "DirectX Diagnostic Tool"의 약자로 DirectX 기능을 테스트하고 비디오 또는 사운드 관련 하드웨어 문제를 해결하기 위해 사용한다. dxdiag를 실행하면 윈도우 버전, CPU와 메모리, DirectX의 버전 등 컴퓨터의 주요 정보를 볼수 있습니다.

msconfig : 윈도우 시동 절차의 문제를 해결하는 데 사용하는 유틸리티 소프트웨어이다. 어느 프로그램이 시작 프로그램에서 실행되는지 수정하고 특정한 설정 파일을 편집하며 윈도우 서비스를 통해 제어를 단순화한다.

regedit : 레지스트리(하드웨어, 소프트웨어, 사용자 그리고 PC나 네트워크의 특성들을 나타내는 값들을 저장)를 편집한다.

mstsc : "Microsoft Terminal Services Client"의 약자로 원격데스크탑 연결을 위한 클라이언트 응용 프로그램이다.

20. <보기>는 0 ~ 199번의 200개 트랙으로 이루어진 디스크 시스템에서, 큐에 저장된 일련의 입출력 요청들과 어떤 디스크 스케줄링(disk scheduling) 방식에 의해 처리된 서비스 순서이다. 이 디스크 스케줄링 방식은 무엇인가? 단, <보기>의 숫자는 입출력할 디스크 블록들이 위치한 트랙 번호를 의미하며, 현재 디스크 헤드의 위치는 트랙 50번이라고 가정한다.

<보기>

○ 요청 큐 : 99, 182, 35, 121, 12, 125, 64, 66

○ 서비스 순서 : 64, 66, 99, 121, 125, 182, 12, 35

- ① FCFS
- ② C-SCAN
- ③ SSTF
- ④ SCAN

정답 체크 :

(2) C-SCAN : 스캔 스케줄링처럼 헤드는 한쪽 방향으로 이동하면서 요청을 처리 하지만, 한쪽 끝에 다다르면 역방향으로 헤드를 이동하는 것이 아니라 다시 처음부터 요청 처리한다. 그러므로 서비스 순서는 64, 66, 99, 121, 125, 182, 12, 35가 된다.

오답 체크 :

(1) FCFS : 요청이 도착한 순서에 따라 처리하므로 서비스 순서는 99, 182, 35, 121, 12, 125, 64, 66가 된다.

(3) SSTF : 디스크 요청을 처리하려고 헤드가 먼 곳까지 이동하기 전에, 현재 헤드 위치에 가까운 모든 요구를 먼저 처리하는 방법이다. 그러므로 서비스 순서는 64, 66, 35, 12, 99, 121, 125, 182가 된다.

(4) SCAN : 입출력 헤드가 디스크의 한쪽 끝에서 다른 끝으로 이동하며, 한쪽 끝에 도달했을 때는 역방향으로 이동하면서 요청한 트랙을 처리한다. 그러므로 서비스 순서는 64, 66, 99, 121, 125, 182, 35, 12가 된다.