2018-계리직-컴퓨터일반-A형-해설-곽후근

1. 다음에서 설명하는 입·출력 장치로 옳은 것은?

ㅇ 중앙처리장치로부터 입·출력을 지시받은 후에는 자신의 명령어를 실행시켜 입·출력을 수행하는 독립된 프로세서이다. ㅇ 하나의 명령어에 의해 여러 개의 블록을 입·출력할 수 있다.

① 버스(Bus)
② 채널(Channel)
③ 스풀링(Spooling)
④ DMA(Direct Memory Access)

정답 체크 :
(2) 채널 : IOP라고 불리며 입출력을 위한 별도의 프로세서(CPU)를 말한다. 입출력 명령어를 처리할 수 있으며 버스 사용권을 가진다. 여러 개의 블록을 입출력할 수 있다.
오답 체크 :
(1) 버스 : 컴퓨터 시스템을 구성하는 장치들 사이에 주소(단방향), 데이터(양방향), 제어(양방향)를 전송하는 데 사용되는 선들의 집합이다. CPU 내부 버스, 시스템 버스, 입출력 버스 등이 존재한다.
(3) 스풀링 : CPU와 입출력 장치(프린터)의 속도 차이를 줄이기 위해 보조기억장치(HDD or SSD)의 일부를 버퍼처럼 사용한다.
(4) DMA : 대용량의 데이터를 이동시킬 때 효과적인 기술로, 기억장치와 입출력 모듈 간의 데이터 전송을 별도의 하드웨어인 DMA 제어기가 처리하고, 중앙처리장치는 개입하지 않도록 하는 입출력 처리 기법이다. CPU로부터 버스의 사용을 허가받아야 한다. 1개의 블록을 입출력할 수 있다.

2. 고객계좌 테이블에서 잔고가 100,000원에서 3,000,000원 사이인 고객들의 등급을 '우대고객'으로 변경하고자 〈보기〉와 같은 SQL문을 작성하였다. ㉠과 ㉡의 내용으로 옳은 것은?

〈보기〉 UPDATE 고객계좌 (㉠) 등급 = '우대고객' WHERE 잔고 (㉡) 100000 AND 3000000

	㉠	㉡
①	SET	IN
②	SET	BETWEEN
③	VALUES	IN
④	VALUES	BETWEEN

정답 체크 :
(2)
SET : UPDATE 문에 사용된다. UPDATE 테이블_이름 SET 속성_이름1 = 값1, 속성_이름2 = 값2, ... [WHERE 조건];
BETWEEN : UPDATE SET WHERE BETWEEN 형태로 사용된다. 예를 들어, 잔고가 100,000원에서 3,000,000원 사이인 고객을 찾을 때 사용한다.
오답 체크 :
(1), (3), (4)

IN : UPDATE SET WHERE IN (SELECT) 형태로 사용된다. 예를 들어, 고객 테이블에서 정소화 고객의 고객아이디를 검색 후 이를 이용할 때 사용한다.
VALUES : INSERT 문에 사용된다. INSERT INTO 테이블_이름[(속성_리스트)] VALUES (속성값_리스트);

3. 네트워크 장치에 대한 설명으로 옳지 않은 것은?
① 허브(Hub)는 여러 대의 단말 장치가 하나의 근거리 통신망(LAN)에 접속할 수 있도록 지원하는 중계 장치이다.
② 리피터(Repeater)는 물리 계층(Physical Layer)에서 동작하며 전송 신호를 재생·중계해 주는 증폭 장치이다.
③ 브리지(Bridge)는 데이터 링크 계층(Data Link Layer)에서 동작 하며 같은 MAC 프로토콜(Protocol)을 사용하는 근거리 통신망 사이를 연결하는 통신 장치이다.
④ 게이트웨이(Gateway)는 네트워크 계층(Network Layer)에서 동작 하며 동일 전송 프로토콜을 사용하는 분리된 2개 이상의 네트워크를 연결해주는 통신 장치이다.
정답 체크 :
(4) 게이트웨이 : OSI 7 Layer 전 계층에서 동작하며, 동일하지 않은 전송 프로토콜을 사용하는 분리된 2개 이상의 네트워크를 연결해준다.
오답 체크 :
(1) 허브 : 2계층 정보를 이용하는 L2 스위치이다. 더미 허브는 패킷을 받으면 다른 모든 포트들로 브로드캐스팅을 하고, 스위칭 허브는 학습된 포트(해당 MAC이 있는 포트)로 보낸다.
(2) 리피터 : 1계층 정보를 이용하는 L1 스위치이다.
(3) 브리지 : 2계층 정보를 이용하는 L2 스위치이다.
Tip!
4계층 정보를 이용하는 L4 스위치에는 부하 분산기, 방화벽 등이 있고, 7계층 정보를 이용하는 L7 스위치에는 어플리케이션 스위치, IDS, IPS, DPI 등이 있다.

4. ㉠에 들어갈 용어로 옳은 것은?

(㉠) (은)는 유사한 문제를 해결하기 위해 설계들을 분류하고 각 문제 유형별로 가장 적합한 설계를 일반화하여 체계적으로 정리해 놓은 것으로 소프트웨어 개발에서 효율성과 재사용성을 높일 수 있다.

① 디자인 패턴
② 요구사항 정의서
③ 소프트웨어 개발 생명주기
④ 소프트웨어 프로세스 모델
정답 체크 :
(1) 디자인 패턴 : 여러 가지 문제에 대한 설계 사례를 분석하여 서로 비슷한 문제를 해결하기 위한 설계들을 분류하고, 각 문제 유형별로 가장 적합한 설계를 일반화해 패턴으로 정립한 것을 의미한다. 소프트웨어 설계에 대한 지식이나 노하우가 문제 유형별로 잘 구체화되어 있을 뿐 아니라, 동일한 문제 유형에 대해서는 그 해결 방법에 대한 지식이나 노하우가 패턴 형태로 충분히 일반화된 것이다.
오답 체크 :
(2) 요구사항 정의서 : 사전적 의미는 이용자가 어떤 문제를 풀거나 목표를 달성하기 위해 필요한

조건이나 능력을 의미하고, 소프트웨어 개발적 의미는 사용자와 개발자 간에 합의한 개발 범위에서 시스템이 제공해야 하는 기능을 의미한다.
(3) 소프트웨어 개발 생명주기(SDLC) : 계획 단계에서 유지보수 단계에 이르기까지 일어나는 일련의 과정을 나타낸다.
(4) 소프트웨어 프로세스 모델 : 소프트웨어를 만들기 위한 단계적인 작업의 틀을 정의한 것이다. 무엇을 하는가에 중점을 두고 결과물 표현에 대한 언급이 없다. 각 단계가 다른 방법론(프로세스의 구체적인 구현이고 어떻게 하는가에 중점을 둔다)으로도 실현이 가능하다. 대표적인 모델로는 폭포수, 나선형, 프로토타이핑, 통합, 애자일 등이 존재한다.

5. 결합도(Coupling)는 모듈 간의 상호 의존 정도 또는 모듈 간의 연관 관계를 의미한다. 아래에 나타낸 결합도를 약한 정도에서 강한 정도 순으로 올바르게 나열한 것은?

ㄱ. 내용 결합도(Content Coupling)
ㄴ. 제어 결합도(Control Coupling)
ㄷ. 자료 결합도(Data Coupling)
ㄹ. 공통 결합도(Common Coupling)

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

정답 체크 :
(1)
자료 결합도 : 모듈들이 매개변수를 통해 데이터만 주고받음으로써 서로 간섭을 최소화하는 관계이고 모듈 간의 독립성 보장한다.
제어 결합도 : 제어 플래그(flag)를 매개변수로 사용하여 간섭하는 관계로서 호출하는 모듈이 호출되는 모듈의 내부 구조를 잘 알고 논리적 흐름을 변경하는 관계이다.
공통 결합도 : 모듈들이 공통 변수(전역변수)를 같이 사용하여 발생하는 관계로서 변수 값이 변하면 모든 모듈이 함께 영향을 받는다.
내용 결합도 : 모듈 간에 인터페이스를 사용하지 않고 직접 왔다 갔다 하는 경우의 관계로서 C 언어의 goto 문을 예로 들 수 있다.
Tip! : 이외에도 자료와 제어 사이에 스탬프 결합이 존재한다.
스탬프 결합도 : 두 모듈 사이에서 정보를 교환할 때 필요한 데이터만 주고받을 수 없고 스탬프처럼 필요 없는 데이터까지 전체를 주고받아야 하는 경우이다. 예를 들면, 레코드나 배열 같은 데이터 구조, C 언어의 구조체(struct)를 들 수 있다.

6. 컴퓨터 알고리즘에 대한 설명으로 옳지 않은 것을 〈보기〉에서 모두 고른 것은?

〈보기〉
ㄱ. 힙 정렬(Heap Sort) 알고리즘의 시간 복잡도는 $O(n^2)$이다.
ㄴ. 0/1 배낭(0/1 Knapsack) 문제에 대하여 다항시간(Polynomial time) 내에 해결 가능한 알고리즘이 개발되었다.
ㄷ. 모든 NP(Non-deterministic Polynomial time) 문제는 컴퓨터를 이용하여 다항시간에 해

결할 수 없다.

① ㄱ

② ㄱ, ㄴ

③ ㄴ, ㄷ

④ ㄱ, ㄴ, ㄷ

정답 체크 :

(4)

ㄱ. 힙 정렬 알고리즘의 시간 복잡도는 O(nlogn)이다.

ㄴ. 배낭 문제란 한 여행가가 가지고 가는 배낭에 담을 수 있는 무게의 최댓값이 정해져 있고, 일정 가치와 무게가 있는 짐들을 배낭에 넣을 때, 가치의 합이 최대가 되도록 짐을 고르는 방법을 찾는 문제이다. 이 배낭문제는 짐을 쪼갤 수 있는 경우(무게가 소수일 수 있는 경우)와 짐을 쪼갤 수 없는 경우(이 경우 짐의 무게는 0 이상의 정수만 가능) 두 가지로 나눌 수 있는데, 짐을 쪼갤 수 있는 경우의 배낭문제를 분할가능 배낭문제(Fractional Knapsack Problem), 짐을 쪼갤 수 없는 경우의 배낭문제를 0-1 배낭문제(0-1 Knapsack Problem)라 부른다. 이 문제는 쪼갤 수 있는 경우에는 그리디 알고리즘으로 다항 시간(어떠한 문제를 계산하는 데에 걸리는 시간 m(n)이 문제의 크기 n의 다항식 함수(nk, k는 문제에 따른 상수값)보다 크지 않은 것)에, 쪼갤 수 없는 경우에는 동적계획법(Dynamic Programming)등으로 의사 다항 시간(다항 시간에 풀 수 없음)에 풀 수 있다. 여기서, 다항 시간이란 일반적으로 빠르게 계산되거나 다루기 쉬운 것을 의미한다.

ㄷ. P 문제는 결정 문제들 중에서 쉽게 풀리는 것을 모아 놓은 집합이다. 어떤 결정 문제가 주어졌을 때, 다항식(Polynomial) 시간 이내에 그 문제의 답을 YES와 NO 중의 하나로 계산해낼 수 있는 알고리즘이 존재한다면, 그 문제는 P 문제에 해당된다. NP 문제는 결정 문제들 중에서 적어도 검산은 쉽게 할 수 있는 것을 모아 놓은 집합으로도 정의할 수 있다. 정확히 말하면, 어떤 결정 문제의 답이 YES일 때, 그 문제의 답이 YES라는 것을 입증하는 힌트가 주어지면, 그 힌트를 사용해서 그 문제의 답이 정말로 YES라는 것을 다항식 시간 이내에 확인할 수 있는 문제가 바로 NP 문제에 해당된다. NP에는 NP-hard와 NP-complete가 존재한다. NP-hard는 모든 경우의 수를 전부 확인해 보는 방법 이외에는 정확한 답을 구할 수 있는 방법이 없는 문제들을 뜻한다. NP-hard는 NP에 속하는 모든 판정 문제를 다항 시간에 다대일 환산할 수 있는 문제들의 집합이므로 다항 시간내에 해결이 가능하다. NP-complete는 NP-hard이지만 다항 시간 내에 해결할 수 없는 문제를 나타낸다. 예를 들어, Travelling Salesman Problem 등을 들 수 있다. NP-complete 문제를 해결하기 위해 근사 알고리즘을 사용한다.

Tip! :

그리디 알고리즘, 동적계획법, 근사 알고리즘은 20번 문제를 참고한다.

7. JAVA 프로그램의 실행 결과로 옳은 것은?

```
class Test {
   public static void main(String[] args) {
   int a = 101;
   System.out.println((a)〉〉2) 〈〈 3);
}
```

① 0

② 200

③ 404
④ 600
정답 체크 :
(2)
(a)〉2) 〈〈 3 : 오른쪽으로 2비트 이동하고, 다시 왼쪽으로 3비트 이동하면 결국 왼쪽으로 1비트 이동한 것이다. 그리고 왼쪽으로 이동하면 주어진 수에 x2를 한 효과를 가지므로 답은 계산하지 않아도 200이 된다.

8. 암호 방식에 대한 설명으로 옳은 것을 〈보기〉에서 모두 고른 것은?

〈보기〉
ㄱ. 대칭키 암호 방식(Symmetric Key Cryptosystem)은 암호화 키와 복호화 키가 동일하다.
ㄴ. 공개키 암호 방식(Public Key Cryptosystem)은 사용자 수가 증가하면 관리해야 할 키의 수가 증가하여 키 변화의 빈도가 높다.
ㄷ. 대칭키 암호 방식은 공개키 암호 방식에 비하여 암호화 속도가 빠르다.
ㄹ. 공개키 암호 방식은 송신자와 발신자가 서로 같은 키를 사용하여 통신을 수행한다.

① ㄱ, ㄴ
② ㄱ, ㄷ
③ ㄴ, ㄷ
④ ㄴ, ㄹ
정답 체크 :
(2)
ㄱ. 대칭키 암호 방식은 암호화 키와 복호화 키가 동일하고, 공개키 암호 방식은 암호화 키와 복호화 키가 다르다.
ㄷ. 대칭키 암호 방식은 수학적 계산이 없기 때문에 공개키 암호 방식에 비하여 암호화 속도가 빠르다.
오답 체크 :
(1), (3), (4)
ㄴ. 해당 설명은 대칭키 암호 방식에 대한 설명이다. 공개키 암호 방식도 관리해야 할 키의 수가 증가하지만 키 변화의 빈도가 높지는 않다.
ㄹ. 해당 설명은 대칭키 암호 방식에 대한 설명이고, 공개키 암호 방식은 송신자와 발신자가 서로 다른 키를 사용하여 통신을 수행한다.

9. 학생 테이블에 튜플들이 아래와 같이 저장되어 있을 때, 〈NULL, '김영희', '서울'〉 튜플을 삽입하고자 한다. 해당 연산에 대한 [결과]와 [원인]으로 옳은 것은? (단, 학생 테이블의 기본 키는 학번이다.)

학번	이름	주소
1	김철희	경기
2	이철수	천안
3	박민수	제주

[결과] [원인]

① 삽입 가능 - 무결성 제약조건 만족
② 삽입 불가 - 관계 무결성 위반
③ 삽입 불가 - 개체 무결성 위반
④ 삽입 불가 - 참조 무결성 위반

정답 체크 :

(3) 개체 무결성 위반 : 기본키에는 NULL을 넣을 수 없다. 만약, NULL을 넣게 되면 투플 사이에 중복이 발생하게 된다. 중복이 발생하게 되면 데이터베이로서의 의미가 존재하지 않는다.

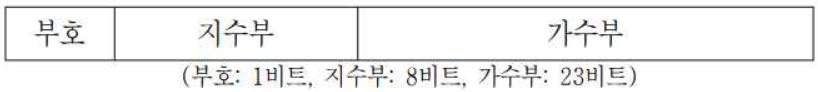
오답 체크 :

(1) 무결성 제약조건 만족 : 개체 무결성 위반으로 무결성 제약조건을 만족하지 않는다.

(2) 관계 무결성 위반 : 어떤 한 투플이 릴레이션에 삽입될 수 있는지 또는 한 릴레이션과 다른 릴레이션의 투플들 간의 관계가 적합한지를 지정한 규정한다. 해당 문제의 답에는 해당되지 않는다.

(4) 참조 무결성 위반 : 외래키는 참조할 수 없는 값을 가질 수 없는 규칙이다. NULL은 가질 수 있지만 참조하지 않은 값을 가질 수는 없다. 해당 문제의 답에는 해당되지 않는다.

10. 10진수 -2.75를 아래와 같이 IEEE 754 표준에 따른 32비트 단정도 부동소수점(Single Precision Floating Point) 표현 방식에 따라 2진수로 표기했을 때 옳은 것은?



(부호: 1비트, 지수부: 8비트, 가수부: 23비트)

① 1000 0000 0000 0000 0000 0000 0000 1011
② 1000 0000 1011 0000 0000 0000 0000 0000
③ 1010 0000 0110 0000 0000 0000 0000 0000
④ 1100 0000 0011 0000 0000 0000 0000 0000

정답 체크 :

(4)

$-2.75_{(10)} = -1.011 \times 2^{1}{}_{(2)}$

부호 : 1

지수 : 1 + 01111111 = 10000000 (바이어스로 127을 사용)

가수 : 01100000000000000000000 (맨 왼쪽은 항상 1이므로 생략)

11. ㉠에 들어갈 용어로 옳은 것은?

주기억장치의 물리적 크기의 한계를 해결하기 위한 기법으로 주기억장치의 크기에 상관없이 프로그램이 메모리의 주소를 논리 적인 관점에서 참조할 수 있도록 하는 것을 (㉠)라고 한다.

① 레지스터(Register)
② 정적 메모리(Static Memory)
③ 가상 메모리(Virtual Memory)
④ 플래시 메모리(Flash Memory)

정답 체크 :

(3) 가상 메모리 : 사용자와 논리적 주소를 물리적으로 분리하여 사용자가 메인 메모리 용량을 초과한 프로세스에 주소를 지정해서 메모리를 제한 없이 사용할 수 있도록 하는 것이다. 프로그램 전체를 동시에 실행하지 않으므로, 요구한 메모리 전체가 아닌 일부만 적재해도 실행 가능하다.

오답 체크 :
(1) 레지스터 : CPU 내부에서 주소, 데이터, 명령어를 저장하는 고속의 임시 저장장치이다.
(2) 정적 메모리 : 배열 처럼 컴파일할 때 메모리를 미리 할당하는 것을 의미한다. 이와 반대로 의미로 동적 메모리(코드를 실행할 때 원하는 메모리를 할당)가 있다.
(4) 플래시 메모리 : EEPROM(전기로 지운후 다시 사용)의 한 종류이지만 EEPROM과는 다르게 블록 단위로 데이터를 입력한다. 읽기와 쓰기 동작이 자유로운 편이어서 RAM과 ROM의 중간적인 위치이다. NAND와 NOR 플래시가 있으며 수명이 제한된다.

12. C 프로그램의 실행 결과로 옳은 것은?

```
#include 〈stdio.h〉
int main( ) {
   int i, sum=0;
   for(i=1; i〈=10; i+=2) {
      if(i%2 && i%3) continue;
      sum += i;
   }
   printf("%d₩n", sum); return 0;
}
```

① 6
② 12
③ 25
④ 55

정답 체크 :
(2)
for(i=1; i〈=10; i+=2) // 주어진 조건에 의해 1, 3, 5, 7, 9에 대해서만 고려한다.
if(i%2 && i%3) continue; // 2 혹은 3으로 나누어 떨어지지 않으면(1, 5, 7) 아래 문장을 실행하지 않고 for 문으로 이동한다.
sum += i; // 2 혹은 3으로 나누어 떨어지는 3, 9를 더한다. 그러면 최종 결과는 12가 된다.

13. 다음에서 설명하는 소프트웨어 개발 방법론으로 옳은 것은?

> 프로세스와 도구 중심이 아닌 개발 과정의 소통을 중요하게 생각 하는 소프트웨어 개발 방법론으로 반복적인 개발을 통한 잦은 출시를 목표로 한다.

① 애자일 개발 방법론
② 구조적 개발 방법론
③ 객체지향 개발 방법론
④ 컴포넌트 기반 개발 방법론

정답 체크 :
(1) 애자일 : 점증적인 프로세스를 채택해서 짧은 반복 주기를 반복하며 점증적으로 자주 출시한다. XP(개발 문서 보다는 소스코드를, 조직적인 개발의 움직임 보다는 개개인의 책임과 용기에 중점을 두는 경향), 스크럼(30일의 개발 주기), 기능 중심 개발(기능 마다 2주 정도의 반복 개발을 실시)이 있다.

오답 체크 :
(2) 구조적 : 분리와 정복(divide and conquer) 원리 적용한다. 구현 언어로 예를 들면 C언어가 있다.
(3) 객체지향 : 객체지향 특성(캡슐화, 다형성, 상속) 원리에 기반한다. 구현 언어로 예를 들면 C++, Java가 있다.
(4) 컴포넌트 기반 : 기존의 시스템이나 소프트웨어를 구성하는 컴포넌트를 조립해서 하나의 새로운 응용 프로그램을 만드는 소프트웨어 개발방법론이다. 기업들은 쇼핑바구니, 사용자 인증, 검색엔진, 카탈로그 등 상업적으로 이용 가능한 컴포넌트를 결합하여 그들의 전자상거래 응용 프로그램을 개발하는 컴포넌트 기반 개발을 사용한다.

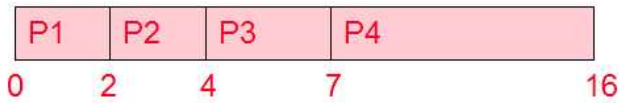
14. 불 대수(Boolean Algebra)에 대한 최소화로 옳지 않은 것은?
① A(A+B) = A
② A+A'B = A+B
③ A(A'+B) = AB
④ AB+AB'+A'B = A
정답 체크 :
(4) AB+AB'+A'B = A(B+B')+A'B = A1+A'B = A+A'B = (A+A')(A+B) = 1(A+B) = A+B (제2분배 법칙 적용)
오답 체크 :
(1) A(A+B) = AA+AB = A+AB = A(1+B)= A1 = A
(2) A+A'B = (A+A')(A+B)= 1(A+B) = A+B (제2분배 법칙 적용)
(3) A(A'+B) = AA'+AB = 0+AB = AB

15. 배열(Array)과 연결리스트(Linked List)에 대한 설명으로 옳지 않은 것은?
① 연결리스트는 배열에 비하여 희소행렬을 표현하는데 비효율적이다.
② 연결리스트에 비하여 배열은 원소를 임의의 위치에 삽입하는 비용이 크다.
③ 연결리스트에 비하여 배열은 임의의 위치에 있는 원소를 접근할 때 효율적이다.
④ n 개의 원소를 관리할 때, 연결리스트가 n 크기의 배열보다 메모리 사용량이 더 크다.
정답 체크 :
(1) 배열의 경우 희소행렬(요소가 0인 항이 대다수인 행렬)을 표현할 때 낭비가 발생하지만, 연결리스트는 낭비가 발생하지 않는다.
오답 체크 :
(2) 배열은 임의의 위치에 삽입할 경우 배열의 요소들을 이동해야 하기 때문에 비용이 발생한다.
(3) 배열은 인덱스를 기반으로 하기 때문에(A[i]) 임의의 위치에 있는 원소를 접근할 때 효율적이다.
(4) 배열은 연속된 메모리로 해당 요소의 크기만 저장하면 되지만, 연결리스트는 비연속된 메모리로 해당 요소의 크기와 다음 요소의 위치를 같이 저장해야 하기 때문에 메모리 사용량이 더 크다.

16. 프로세스 P1, P2, P3, P4를 선입선출(First In First Out) 방식으로 스케줄링을 수행할 경우 평균응답시간으로 옳은 것은? (단, 응답시간은 프로세스 도착시간부터 처리가 종료될 때까지의 시간을 말한다)

프로세스	도착시간	처리시간
P1	0	2
P2	2	2
P3	3	3
P4	4	9

① 3
② 4
③ 5
④ 6
정답 체크 :
(3)
주어진 조건으로 간트 차트를 그리면 다음과 같다.



각 프로세스의 응답 시간을 계산하면 다음과 같다.
P1 = 2 - 0 = 2
P2 = 4 - 2 = 2
P3 = 7 - 3 = 4
P4 = 16 - 4 = 12
평균응답시간은 다음과 같다.
평균응답시간 = (P1+P2+P3+P4) / 4 = 5

문 17. TCP/IP 프로토콜에 대한 설명으로 옳은 것은?
① TCP는 비연결형 프로토콜 방식을 사용한다.
② TCP는 네트워크 계층(Network Layer)에 속한다.
③ IP는 잘못 전송된 패킷에 대하여 재전송을 요청하는 기능을 제공한다.
④ IP는 각 패킷의 주소 부분을 처리하여 패킷이 목적지에 도달할 수 있도록 한다.
정답 체크 :
(4) IP는 전체 메시지 중에 1개의 패킷을 목적지에 배달한다.
오답 체크 :
(1) TCP는 연결형 프로토콜 방식을 사용한다.
(2) TCP는 전송 계층(Transport Layer)에 속한다.
(3) IP는 잘못 전송된 패킷에 대하여 재전송을 요청하는 기능을 제공하지 않는다. (비연결형, 비신뢰성)

문 18. 다음에서 설명하는 용어로 가장 옳은 것은?
프랭크 로젠블라트(Frank Rosenblatt)가 고안한 것으로 인공신경망 및 딥러닝의 기반이 되는 알고리즘이다.
① 빠른 정렬(Quick Sort)
② 맵리듀스(MapReduce)

③ 퍼셉트론(Perceptron)
④ 디지털 포렌식(Digital Forensics)
정답 체크 :
(3) 퍼셉트론 : 인공신경망의 한 종류로서, 1957년에 코넬 항공 연구소(Cornell Aeronautical Lab)의 프랑크 로젠블라트 (Frank Rosenblatt)에 의해 고안되었다. 이것은 가장 간단한 형태의 피드포워드(Feedforward) 네트워크(선형분류기)로도 볼 수 있다. 퍼셉트론이 동작하는 방식은 다음과 같다. 각 노드의 가중치와 입력치를 곱한 것을 모두 합한 값이 활성함수에 의해 판단되는데, 그 값이 임계치(보통 0)보다 크면 뉴런이 활성화되고 결과값으로 1을 출력한다. 뉴런이 활성화되지 않으면 결과값으로 -1을 출력한다(해당 출력으로 결정을 수행). 단층 퍼셉트론은 XOR 연산이 불가능하지만, 다층 퍼셉트론으로는 XOR 연산이 가능하다.
오답 체크 :
(1) 빠른(퀵) 정렬 : 분할정복기법으로 정렬을 수행한다.
(2) 맵리듀스 : 빅데이터를 처리하기 위한 소프트웨어 프레임워크이다. 맵리듀스를 이용하면 저장된 파일 데이터를 분산된 서버의 CPU와 메모리 자원을 이용해 쉽고 빠르게 분석할 수 있다.
(4) 디지털 포렌식 : 전자적 증거물 등을 사법기관에 제출하기 위해 데이터를 수집, 분석, 보고서를 작성하는 일련의 작업을 말한다. 사이버 해킹 공격, 사이버 범죄시 범죄자들이 컴퓨터, 이메일, IT 기기, 스마트폰 등의 운영체제, 애플리케이션, 메모리 등에 다양한 전자적 증거를 남기게 되면서, 사이버 범죄자 추적 및 조사에 핵심적인 요소가 되고 있다.

문 19. 관계형 데이터베이스의 뷰(View)에 대한 장점으로 옳지 않은 것은?
① 뷰는 데이터의 논리적 독립성을 일정 부분 제공할 수 있다.
② 뷰를 통해 데이터의 접근을 제어함으로써 보안을 제공할 수 있다.
③ 뷰에 대한 연산의 제약이 없어서 효율적인 응용프로그램의 개발이 가능하다.
④ 뷰는 여러 사용자의 상이한 응용이나 요구를 지원할 수 있어서 데이터 관리를 단순하게 한다.
정답 체크 :
(3) 뷰에 대한 삽입/수정/삭제 연산은 실제로 기본 테이블에 수행되므로 결과적으로 기본 테이블의 변경된다. 그러므로 뷰에 대한 삽입/수정/삭제 연산은 제한적으로 수행된다. 예를 들어, 기본 테이블의 기본키를 구성하는 속성이 포함되어 있지 않은 뷰는 연산을 수행할 수 없다.
오답 체크 :
(1) 뷰는 물리적인 테이블을 기반으로 만들어진 논리적인 테이블로서 데이터의 논리적 독립성을 일정 부분 제공할 수 있다.
(2) 자신에게 제공된 뷰를 통해서만 데이터에 접근하도록 권할 설정이 가능하므로 데이터의 보안 유지에 도움이 된다.
(4) 여러 사용자의 상이한 응용이나 요구에 해당하는 뷰만 제공하면 되기 때문에 데이터 관리를 단순하게 할 수 있다. 즉, 제공된 뷰와 관련이 없는 다른 내용에 대해서는 신경쓸 필요가 없다.

문 20. 다음에서 설명하는 알고리즘 설계 기법으로 가장 알맞은 것은?
해결하고자 하는 문제의 최적해(Optimal Solution)가 부분 문제들의 최적해들로 구성되어 있을 경우, 이를 이용하여 문제의 최적해를 구하는 기법이다.
① 동적 계획법(Dynamic Programming)
② 탐욕적 알고리즘(Greedy Algorithm)

③ 재귀 프로그래밍(Recursive Programming)
④ 근사 알고리즘(Approximation Algorithm)
정답 체크 :
(1) 동적 계획법 : 일반적으로 주어진 문제를 풀기 위해서, 문제를 여러 개의 하위 문제(subproblem)로 나누어 푼 다음, 그것을 결합하여 최종적인 목적에 도달하는 것이다. 각 하위 문제의 해결을 계산한 뒤, 그 해결책을 저장하여 후에 같은 하위 문제가 나왔을 경우 그것을 간단하게 해결할 수 있다. 이러한 방법으로 동적 계획법은 계산 횟수를 줄일 수 있다. 특히 이 방법은 하위 문제의 수가 기하급수적으로 증가할 때 유용하다. 예를 들어, 최단 경로 문제 혹은 행렬의 제곱 문제 등에 유용하다. (최적해를 구한다는 것이 포인트)
오답 체크 :
(2) 탐욕적 알고리즘 : 최적해를 구하는 데에 사용되는 근사적인 방법으로, 여러 경우 중 하나를 결정해야 할 때마다 그 순간에 최적이라고 생각되는 것을 선택해 나가는 방식으로 진행하여 최종적인 해답에 도달한다. 순간마다 하는 선택은 그 순간에 대해 지역적으로는 최적이지만, 그 선택들을 계속 수집하여 최종적(전역적)인 해답을 만들었다고 해서, 그것이 최적이라는 보장은 없다. (최적해를 보장하지 않는다라는 것이 포인트)
(3) 재귀 프로그래밍 : 자신을 정의할 때 자기 자신을 재참조하는 방법을 뜻하며, 이를 프로그래밍에 적용한 재귀 호출(Recursive call)의 형태로 많이 사용된다. 또 사진이나 그림 등에서 재귀의 형태를 사용하는 경우도 있다.
(4) 근사 알고리즘 : 어떤 최적화 문제에 대한 해의 근사값을 구하는 알고리즘을 의미한다. 이 알고리즘은 가장 최적화되는 답을 구할 수는 없지만, 비교적 빠른 시간에 계산이 가능하며 어느 정도 보장된 근사해를 계산할 수 있다. 근사 알고리즘은 NP-완전 문제등 현재 알려진 빠른 최적화 알고리즘이 없을 문제에 대해 주로 사용된다. (최적해를 구할 수 없어 근사해를 구한다는 것이 포인트)