2014-국회직-컴퓨터일반-가형-해설-곽후근

1. 관계형 데이터베이스에서 불필요한 정보 중복으로 인한 문제점이 없도록 릴레이션(relation)을 작게 분해하는 과정을 의미하는 것은?
① 조인(join)
② 인덱싱
③ 정규화(normalization)
④ 증분 백업(incremental backup)
⑤ 스키마 변환
정답 체크 :
(3) 정규화 : 이상 현상이 발생하지 않도록, 릴레이션을 관련 있는 속성들로만 구성하기 위해 릴레이션을 분해(decomposition)하는 과정이다. 함수적 종속성(속성들 간의 관련성)을 판단하여 정규화를 수행한다.
오답 체크 :
(1) 조인 : 조인 속성을 이용해 두 릴레이션을 조합하여 결과 릴레이션을 구성한다. 조인 속성이란 두 릴레이션이 공통으로 가지고 있는 속성을 의미한다.
(2) 인덱싱 : 검색 속도를 높이기 위한 자료 구조이다. 인덱스는 일반적으로 키-필드로 이루어져있고 테이블에 대한 세부 정보를 가지지 않는다. 별도의 세부 정보의 이용 없이 키 값을 기초로 하기 때문에 검색과 정렬속도를 향상시킨다. 인덱싱하고자 하는 컬럼을 키로 하는 index table을 만든다.
(4) 증분 백업 : 선택된 데이터의 Full 백업(데이터를 모두 백업하는 방식) 이후 변경, 추가된 데이터만 백업하는 방식이다. 단, 주기적으로 Full 백업을 수행한다.
(5) 스키마 변환 : 개념적 스키마(E-R 다이어그램)에서 논리적 스키마(릴레이션 스키마)로의 변환과 논리적 스키마에서 물리적 스키마로의 변환이 존재한다.

2. 다음과 같은 정규식(regular expression)이 있다. 아래의 문자열 중 이 정규식에 의하여 만들어질 수 있는 것은?

d+(e|ba)*

① dbb
② ebaba
③ dbea
④ ddebaeba
⑤ ddaa
정답 체크 :
(4)
정규식의 규칙은 다음과 같다.
+ : 앞 문자가 하나 이상이다. (예 : dd)
( ) : 하나의 패턴 구분자 안에 서브 패턴을 지정해서 사용할 경우 괄호로 묶어주는 방식을 사용한다. (예 : (e|ba)*)
| : or를 나타낸다 (예 : eba)
* : 앞 문자가 0개 이상이다. (예 : ebaeba)

이들을 모두 합치면 ddebaeba가 된다.

3. 다음 자바 프로그램의 오류를 수정하는 방법 중 옳은 것은?

```
1  class Node {
2    int val;
3  }
4  public class Example {
5    public static void main(String args[]) {
6      Node n[] = new Node[100];
7      for (int i = 0; i < n.length; i++) {
8        n[i].val = 0;
9      }
10   }
11 }
```

① 7번째 줄을 다음과 같이 수정한다.
for (int i = 0; i < 100; i ++) {
② 7번째 줄 다음에 아래 문장을 추가한다.
n[i] = new Node() ;
③ 6번째 줄을 아래 문장과 같이 수정한다.
Node n[100] = new Node[] ;
④ 8번째 줄을 아래 문장과 같이 수정한다.
n[i]->val = 0;
⑤ Node의 선언 부분을 class Example 내부로 이동한다.
정답 체크 :
(2)
Node n[] = new Node[100]; // 레퍼런스 배열을 생성한다. (C언어에서 2중 포인터(2차원 배열)의 개념과 유사)
n.length; // 레퍼런스 배열이 생성되면 자동으로 주어진다. (C언어에서는 자동으로 주어지지 않음)
n[i] = new Node(); // 배열의 각 원소 객체를 생성한다. (C언에서 포인터(1차원 배열)의 개념과 유사)
Tip! : 자바에서 객체 배열을 만드는 방법은 항상 위와 같으므로 무조건 숙지하는 것이 좋다.

4. 다음 C 프로그램의 출력 결과로 옳은 것은?

```
# include<stdio.h>
int *func(int a, int *x)
{
  a = a + 10;
  x = x + 1;
  *x = *x * 2;
  return x;
```

```
}
int main( )
{
  int i;
  int x = 10;
  int *p;
  int a[100];
  for(i = 0; i 〈 100; i++) a[i] = i * 10;
  p = func(x, a);
  printf("sum = %d₩n", x + a[0] + a[1] + p[0] + p[1]);
}
```

① 오류가 발생한다.
② 60
③ 61
④ 70
⑤ 80
정답 체크 :
(4)
a[0] = 0 * 10 = 0, a[1] = 1 * 10 = 10, a[2] = 2 * 10 = 20; // for 문을 통해 a 배열에 값이 할당된다. 규칙적이므로 한두 개만 해보면 된다.
p = func(x, a); // x는 call-by-value이므로 값(10)이 바뀌지 않는다(함수 내에서 관심을 주지 않아도 된다). a는 call-by-reference이므로 주의 깊게 봐야 하고, 반환 되는 주소도 주의 깊게 봐야 한다. a가 함수 내에서 x에 대입되므로 헷갈리지 말아야 한다.
x = x + 1; // x의 주소가 하나 증가한다. x[1]이 된다.
x[1] = x[1] * 2 = 20; // x[1]은 10이기 때문에 최종 결과는 20이 된다. (즉, a[1]이 20이 된다.)
return x; // x[1]의 주소를 반환한다. (즉, a[1]의 주소를 반환한다.)
p = a[1] = p[0] = 20; // p는 a[1]의 주소를 가지게 되고, p[0]는 a[1]의 값인 20을 가지게 된다.
p[1] = 20 = a[2]; // p[1]은 a[2]가 되므로 20을 가지게 된다.
x + a[0] + a[1] + p[0] + p[1] = 10 + 0 + 20 + 20 + 20 = 70; // 이들을 다 더하면 70이 된다.

5. 이미 정렬되어 있는 목록에 새로운 데이터를 입력하였을 때 가장 빠르게 정렬 결과를 얻을 수 있는 것은?
① 기수 정렬(radix sort)
② 퀵 정렬(quick sort)
③ 삽입 정렬(insertion sort)
④ 힙 정렬(heap sort)
⑤ 합병 정렬(merge sort)
정답 체크 :

(3) 삽입 정렬 : 새로운 데이터를 n번만 비교하면 된다. 최선의 시간 복잡도는 O(n)이 된다.
오답 체크 :
(1) 기수 정렬 : 새로운 데이터를 다시 정렬해야 한다. 최선의 시간 복잡도는 O(dn)이 된다.
(2) 퀵 정렬 : 새로운 데이터를 다시 정렬해야 한다. 최선의 시간 복잡도는 $O(n\log_2 n)$이 된다.
(4) 힙 정렬 : 새로운 데이터를 다시 정렬해야 한다. 최선의 시간 복잡도는 $O(n\log_2 n)$이 된다.
(5) 합병 정렬 : 새로운 데이터를 다시 정렬해야 한다. 최선의 시간 복잡도는 $O(n\log_2 n)$이 된다.
Tip! : 이미 정렬된 경우 시간 복잡도에서 최선의 경우에 해당된다.

6. 선형(linear) 자료구조로만 짝지어진 것은?
① 배열, 리스트, 스택, 큐
② 배열, 트리, 스택, 큐
③ 배열, 리스트, 그래프, 큐
④ 배열, 리스트, 스택, 트리
⑤ 트리, 리스트, 스택, 큐
정답 체크 :
(1)
선형 자료구조(1:1로 연결) : 배열, 연결 리스트, 스택, 큐
오답 체크 :
(2), (3), (4), (5)
비선형 자료구조(1:N 또는 N:N으로 연결) : 트리, 그래프

7. 최대 힙(max heap)에 대하여 올바르게 설명한 것은?
① 두개의 자식 노드를 갖는 노드의 경우 좌측 자식 노드 보다는 크고, 우측 자식 노드보다는 작은 키(key) 값이 저장 된다.
② 힙을 중위 순회(in-order traversal)하면 정렬된 순서로 데이터를 얻을 수 있다.
③ 완전 그래프(complete graph)에 해당된다.
④ 루트 노드는 트리에 저장된 키(key) 값 중 가장 큰 값을 갖는다.
⑤ 형제 노드들의 키(key) 값은 좌에서 우로 정렬된 순서를 유지한다.
정답 체크 :
(4) 루트 노드 : 최대 힙은 부모 노드의 키값이 자식 노드의 키값보다 크거나 같고, 최소 힙은 부모 노드의 키값이 자식 노드의 키값보다 작거나 같다.
오답 체크 :
(1) 두 개의 자식 노드 : 이진 탐색 트리에 해당된다.
(2) 중위 순회 : 이진 탐색 트리에 해당된다.
(3) 완전 그래프 : 완전이진트리이다.
(5) 형제 노드 : 이진 탐색 트리에 해당된다.

8. RAID에 대하여 올바르게 설명한 것은?
① 자기 테이프를 효율적으로 구성하기 위한 기술이다.
② 자기 디스크에 더 많은 양의 데이터를 저장하기 위한 기술이다.
③ 읽기 전용 보조기억 장치를 구성하기 위한 것이다.

④ RAID 레벨 0은 빠르기보다는 데이터의 안정성에 중점을 둔 구성 방법이다.
⑤ RAID 레벨 5는 패리티(parity)가 모든 디스크에 분산된다.
정답 체크 :
(5) RAID 레벨 5 : 패리티 비트를 저장하는 볼륨을 별도로 설치(레벨 4)하지 않고, 데이터를 저장하는 볼륨에 패리티 비트를 분산하여 저장한다.
오답 체크 :
(1) 자기 테이프 : 저렴하고 크기가 작은 여러 개의 독립된 하드 디스크(자기 디스크, HDD)들을 묶어 하나의 기억장치처럼 사용할 수 있는 방식이다.
(2) 더 많은 양의 데이터 : 여러 개의 독립된 디스크들이 일부 중복된 데이터를 나눠서 저장하고 신뢰성(안정성), 성능(속도)을 향상시키는 기술을 의미한다.
(3) 읽기 전용 보조기억장치 : 읽기와 쓰기가 가능한 보조기억장치(HDD)를 이용한다.
(4) RAID 레벨 0 : 2개 이상의 디스크를 사용하여 2개 이상의 볼륨을 구성한 구조로 여분(redundancy) 디스크를 포함하지 않아서 오류 검출 기능은 없다. 높은 신뢰성을 요구하기 보다는 성능과 용량을 중요시하는 시스템에 사용한다.

9. NAND flash 메모리에 대한 설명으로 옳지 않은 것은?
① read와 write가 page 단위로 수행된다.
② erase가 block 단위로 수행 된다.
③ overwrite를 하기 위해서는 erase가 선행되어야 한다.
④ erase 속도가 read보다 빠르다.
⑤ 전원 공급이 끊겨도 데이터를 잃어버리지 않는다.
정답 체크 :
(4) erase 속도 : 물리적은 수행 방법을 떠나 읽기는 페이지 단위로 수행되고 지우기는 블록 단위로 수행되므로 읽기가 더 빠르다.
오답 체크 :
(1) read와 write : 페이지 단위로 읽기/쓰기 동작이 가능하다.
(2) erase : 블록(블록을 나누면 페이지가 된다) 단위로 지워야 한다.
(3) overwrite : 덮어 쓸 수 없으므로, 모든 블록을 지우기 전까지는 해당 자료를 변경할 수 없다.
(5) 전원 공급 : 메모리 칩 안에 정보를 유지시키는 데에 전력이 필요 없는 비휘발성 메모리이다.

10. RISC(Reduced Instruction Set Computer) 방식 컴퓨터에 대한 설명으로 옳지 않은 것은?
① RISC 방식은 CISC(Complex Instruction Set Computer) 방식보다 간단한 명령어 구조를 사용한다.
② RISC 방식은 CISC 방식보다 파이프라이닝 구현이 용이하다.
③ RISC 방식은 CISC 방식보다 주소 지정 방식이 간단하다.
④ RISC 방식은 고정된 길이의 명령어 형식으로 디코딩이 간단하다.
⑤ RISC 방식의 CPU는 CISC 방식보다 상대적으로 적은 수의 레지스터를 사용한다.
정답 체크 :
(5) 레지스터 : 많은 수의 레지스터를 사용한다. CISC보다 제어 장치를 간단하게 만들 수 있기 때문에 남는 돈으로 레지스터와 캐시 등을 추가하였다.
오답 체크 :

(1) 명령어 구조 : CISC에는 많은 명령어가 존재하지만 자주 사용하는 명령어는 몇 개 안된다는 사실에서 출발하였다. 그러므로 CISC에 비해 적은 수의 명령어와 간단한 명령어 구조를 가진다.
(2) 파이프라이닝 : 명령어 구조가 간단하기 때문에 명령어 단계에 들어가는 시간을 일정하게 맞출 수 있고 이는 파이프라이닝의 성능 개선으로 이어진다. 또한 명령어 연산 코드의 해독과 레지스터 오퍼랜드의 액세스가 동시에 일어나는 게 가능하다.
(3) 주소 지정 방식 : 주소 지정방식에 있어서도, 단순하게 레지스터 주소 지정 방식을 사용하므로 적은 수의 간단한 주소지정 방식을 사용 할 수 있다.
(4) 명령어 형식 : 적은 수의 단순한(고정된) 명령어 형식을 사용할 수 있다.

11. 그리드 컴퓨팅 시스템(grid computing system)에 대한 설명으로 옳지 않은 것은?
① PC, 워크스테이션, 서버 등과 같은 다양한 컴퓨터들의 네트워크이다.
② 문제를 여러 조각으로 나누어 개별 컴퓨터에 분할하여 처리하도록 해준다.
③ 한 컴퓨터는 데이터베이스 서버로 지정되고 다른 컴퓨터는 그래픽 처리 전용이 되는 등 각각의 컴퓨터들은 특수한 작업을 수행한다.
④ 크고 복잡한 계산을 할 수 있게 해준다.
⑤ 컴퓨터를 추가하거나 제거하여 쉽게 규모를 조절할 수 있다.
정답 체크 :
(3) 특수한 작업 : 그리드 컴퓨팅에서 그리드상의 자원을 통제하고 할당하려면, 글로버스얼라이언스(그리드 컴퓨팅을 위한 기반 기술을 개발하기 위해 설립된 조직)나 개인 제공자가 제공하는 공개소스 소프트웨어 같은 소프트웨어 프로그램이 필요하다. 클라이언트 소프트웨어는 서버의 응용 프로그램과 통신한다. 이런 서버 소프트웨어는 데이터와 응용 프로그램 코드를 일정 단위로 분할한 뒤, 분할된 코드를 그리드상의 컴퓨터에 배분한다. 즉, 개별 컴퓨터는 특수한 작업을 수행하지 않는다.
오답 체크 :
(1) 다양한 컴퓨터 : 서로 다른 기종(heterogeneous)의 컴퓨터들을 하나로 묶고 이들을 연결하기 위해 별도의 표준 프로토콜이 필요하다.
(2) 여러 조각 : 대용량 데이터에 대한 연산을 작은 소규모 연산들로 나누어 작은 여러 대의 컴퓨터들로 분산시켜 수행한다.
(4) 크고 복잡한 계산 : 고도의 연산 작업(computation intensive jobs) 혹은 대용량 처리(data intensive jobs)를 수행한다.
(5) 규모 조절 : 클러스터 컴퓨팅과 마찬가지로 컴퓨터를 추가하면 그에 따른 성능(규모) 향상을 얻을 수 있다.

12. 추상 클래스(abstract class)에 대하여 올바르게 설명한 것은?
① 상세 클래스(concrete class)라고도 부른다.
② 상속을 하여 파생 클래스를 만들 수 없다.
③ 어떠한 클래스의 파생 클래스는 추상 클래스가 될 수 없다.
④ 추상 클래스의 객체를 직접 생성할 수 없다.
⑤ 데이터 멤버를 포함할 수 없다.
정답 체크 :
(4) 객체 생성 : 추상 클래스는 온전한 클래스가 아니기 때문에 인스턴스(객체)를 생성할 수 없다.
오답 체크 :

(1) 상세 클래스 : 추상 클래스의 반대이다. 객체를 만들기 위해서 만드는 클래스이다.
(2) 상속 : 추상 클래스는 상속의 용도로 많이 사용된다.
(3) 파생 클래스 : 추상 클래스의 파생 클래스는 추상 클래스가 될 수 있다.
(4) 데이터 멤버 : 추상 클래스는 추상 메소드와 데이터 멤버를 포함할 수 있다.

13. 소프트웨어 재사용에 대한 설명으로 옳지 않은 것은?
① 소프트웨어 개발 시간과 비용 절감
② 프로젝트 실패 위험률 감 소
③ 소프트웨어 개발자의 생산성 증가
④ 소프트웨어 구축에 대한 지식 공유
⑤ 새로운 소프트웨어 개발방법론 개발
정답 체크 :
(5) 개발 방법론 개발 : 소프트웨어 재사용은 기존의 코드를 이용하는 것이므로 새로운 소프트웨어 개발 방법론 개발과는 무관하다.
오답 체크 :
(1) 개발 시간과 비용 절감 : 기존의 소프트웨어를 재사용하게 되면 개발 시간을 단축하고 비용을 절감할 수 있다.
(2) 실패 위험률 감소 : 기존의 검증된 코드를 사용하면 실패 위험을 감소시킬 수 있다.
(3) 개발자의 생산성 증가 : 개발자가 모든 코드를 개발할 필요가 없으므로 개발자의 생산성이 증가된다.
(4) 지식 공유 : 자신의 코드를 다른 사람들과 공유할 수 있다.

14. TCP/IP 프로토콜에서 IP 호스트가 자신의 물리 네트워크 주소(MAC)는 알지만 IP 주소를 모르는 경우, 서버에게 IP 주소를 요청하기 위해 사용하는 프로토콜은?
① RARP
② ICMP
③ ARP
④ IGMP
⑤ UDP
정답 체크 :
(1) RARP : MAC 주소(물리 주소)를 IP 주소(논리 주소)로 바꿔준다.
오답 체크 :
(2) ICMP : 인터넷 제어 메시지 프로토콜은 RFC 792에서 정의한 인터넷 프로토콜 모음 중의 하나이다. ICMP 메시지들은 일반적으로 IP 동작에서 진단이나 제어로 사용되거나 오류에 대한 응답으로 만들어진다. 예를 들어, 핑(ping) 유틸리티는 ICMP "에코 요청(Echo request)"과 "에코 응답(Echo reply)" 메시지를 사용해 구현할 수 있다.
(3) ARP : IP 주소(논리 주소)를 MAC 주소(물리 주소)로 바꿔준다.
(4) IGMP : 인터넷 그룹 관리 프로토콜은 호스트 컴퓨터와 인접 라우터가 멀티캐스트 그룹 멤버십을 구성하는 데 사용하는 통신 프로토콜이다. 특히 IPTV와 같은 곳에서 호스트가 특정 그룹에 가입하거나 탈퇴하는데 사용하는 프로토콜을 가리킨다.
(5) UDP : 연결을 설정하지 않고 수신자가 데이터를 받을 준비를 확인하는 단계를 거치지 않고 단

방향으로 정보를 전송한다. UDP를 사용하는 애플리케이션에는 도메인 이름 서비스(DNS), IPTV, 음성 인터넷 프로토콜(VoIP), TFTP, IP 터널, 그리고 많은 온라인 게임 등이 있다.

15. 프로세스의 스케줄링 정책 중 평균 대기시간(average waiting time)이 가장 적은 것은?
① Round Robin
② Shortest Job First
③ First Come First Served
④ SCAN
⑤ C-LOOK
정답 체크 :
(2) SJF : 프로세서(CPU)가 사용 가능할 때 실행 시간이 가장 짧은 작업(프로세스)에 할당하기 때문에 평균대기시간이 감소한다.
오답 체크 :
(1) RR : 프로세스에 정의된 규정 시간량(Time Quantum) 또는 시간 할당량(Time Slice) 만큼 프로세서(CPU)를 제공한다. 작업 길이가 다양할 때는 이전 작업을 마친 후 후보가 규정 시간량을 마치고 다음 작업으로 이동하기 때문에, 평균대기시간이 FCFS보다 적다.
(3) FCFS : 프로세서(CPU)를 요청하는 순서대로 할당하기 때문에 장기 실행 프로세스가 뒤의 프로세스(작업)를 모두 지연시켜 평균대기시간이 길어져 최악의 대기시간이 된다.
(4) SCAN : 프로세스 스케줄링이 아니라 디스크 스케줄링이다.
(5) C-LOOK : 프로세스 스케줄링이 아니라 디스크 스케줄링이다.

16. CPU를 점유하고 있는 프로세스를 교체하기 위해, 이전의 프로세스의 상태를 보관하고 새로 진입하는 프로세스의 상태를 적재하는 작업은?
① 상호배제(mutual exclusion)
② 동기화(synchronization)
③ 교착상태(dead-lock)
④ 스케줄링(scheduling)
⑤ 문맥교환(context switching)
정답 체크 :
(5) 문맥교환 : 실행 중인 프로세스의 제어를 다른 프로세스에 넘겨 실행 상태가 되도록 하는 것이다. 프로세스 문맥 교환이 일어나면 프로세서의 레지스터에 있던 내용 저장한다.
오답 체크 :
(1) 상호배제 : 자원을 최소 하나 이상 비공유한다. 즉, 한 번에 프로세스 하나만 해당 자원 사용할 수 있어야 한다. 사용 중인 자원을 다른 프로세스가 사용하려면, 요청한 자원 해제될 때 까지 대기한다.
(2) 동기화 : 변수나 파일은 프로세스별로 하나씩 차례로 읽거나 쓰도록 해야 하는데, 공유 자원을 동시에 사용하지 못하게 실행을 제어하는 방법 뜻한다.
(3) 교착상태 : 다중 프로그래밍 시스템에서, 프로세스가 결코 일어나지 않을 사건(event)을 기다리는 상태이다. 프로세스가 교착 상태에 빠지면, 작업 정지되어 명령 진행 불가하다.
(4) 스케줄링 : 프로세스 스케줄링은 여러 개의 프로세스가 존재하면 다양한 스케줄링 방법들을 이용해서 프로세서(CPU)에게 프로세스(작업)을 할당하는 것이다. 디스크 스케줄링은 여러 개의 디스크

작업 요청이 존재하면 다양한 스케줄링 방법들을 이용해서 해당 작업들을 처리하는 것이다.

17. 가상(virtual) 기억 장치의 페이지 부재 발생 시, 페이지 교체기법 중 LRU 방식에 대하여 올바르게 설명한 것은?
① 가장 오랫동안 사용되지 않고 있는 페이지를 교체 대상으로 선택한다.
② 참조된 횟수가 가장 적은 페이지를 교체 대상으로 선택한다.
③ 주기억장치에 가장 먼저 적재된 페이지를 교체 대상으로 선택한다.
④ 프로세스에 더 많은 수의 페이지 프레임을 할당하였을 때 오히려 페이지 부재의 발생 횟수가 증가하는 현상이 발생할 수 있다.
⑤ 페이지 부재 발생 비율에 따라 페이지 프레임의 수를 추가 할당하거나 회수하는 기법이다.
정답 체크 :
(1) 가장 오랫동안 사용되지 않고 있는 페이지 : LRU는 프로세스가 가장 최근의 페이지에 액세스했다는 것은 멀지 않아 다시 액세스할 가능성이 있다는 의미이다. 과거 오랫동안 사용하지 않은 페이지로 대치하는 효과로 생각할 수 있다.
오답 체크 :
(2) 참조된 횟수가 가장 적은 페이지 : LFU는 각 페이지마다 참조 횟수 카운터가 있으며, 수가 가장 작은 페이지 대치한다. 프로세스의 초기 단계에서 한 페이지를 많이 사용한 후 다시는 사용하지 않을 때 곤란하다.
(3) 주기억장치에 가장 먼저 적재된 페이지 : FIFO에 대한 설명이다.
(4) 더 많은 수의 페이지 프레임 할당 : FIFO에서 벨래디의 변이(Belady's anomaly)에 대한 설명이다.
(5) 페이지 부재 발생 비율 : 스레싱(Thrashing)을 해결하기 위한 페이지 부재 빈도(Page Fault Frequency) 방법에 대한 설명이다.

18. IPv4와 IPv6에서 IP 주소의 길이는 각각 몇 비트인가?
① IPv4 : 16 비트 , IPv6 : 32 비트
② IPv4 : 16 비트 , IPv6 : 64 비트
③ IPv4 : 32 비트 , IPv6 : 64 비트
④ IPv4 : 32 비트 , IPv6 : 128 비트
⑤ IPv4 : 64 비트 , IPv6 : 256 비트
정답 체크 :
(4)
IPv4 : 32비트 = 4바이트
IPv6 : 128비트 = 16바이트
Tip! : IPv6의 주소를 바이트로 물어보면 헷갈릴 수 있으므로 기억해두는 것이 좋다.

19. 한 종류의 게이트만을 조합하여 모든 다른 게이트로 사용할 수 있는 유니버셜 게이트(universal gate)는?
① AND
② OR
③ XOR

④ NAND
⑤ XNOR
해설)
정답 체크 :
(4)
NAND : Universal Gate로 해당 게이트로 모든 다른 게이트(AND, OR, NOT)를 만들 수 있다. 이에 해당 하는 게이트에는 NOR도 존재한다. 하나를 예로 들면, NAND 게이트 2개를 이용하면 AND 게이트를 만들 수 있다.

20. 다음에서 설명하는 것은 무엇을 말하는 것인가?

기존의 유선통신을 기반으로 한 인터넷이나 모바일 인터넷보다 진화된 단계로 인터넷에 연결된 기기가 사람의 개입 없이 상호 간에 알아서 정보를 주고받아 처리하며, 사물은 물론이고 현실과 가상 세계의 모든 정보와 상호작용하는 개념을 말한다.

① 사물인터넷(Internet of things)
② 클라우드 컴퓨팅(cloud computing)
③ 유틸리티 컴퓨팅(utility computing)
④ 빅데이터 서비스(big data service)
⑤ 딥러닝(deep learning)
정답 체크 :
(1) 사물 인터넷 : 각종 사물에 컴퓨터 칩과 통신 기능을 내장해 인터넷에 연결하는 기술을 의미한다. 사물끼리 정보를 주고받기 위하여 인터넷으로 연결되어 있는 사물 공간 연결망이다.
오답 체크 :
(2) 클라우드 컴퓨팅 : 각종 소프트웨어와 데이터를 인터넷과 연결된 중앙 컴퓨터에 저장한다. 필요할 때마다 컴퓨터나 스마트폰 같은 단말기로 접속하여 데이터를 내려받아 사용하고 다시 업로드하는 방식이다.
(3) 유틸리티 컴퓨팅 : 유틸리티(수도, 전기 등)처럼 사용한 만큼 돈을 내자는 개념이다. 대표적인 예로 클라우드 컴퓨팅이 있다.
(4) 빅 데이터 서비스 : 굉장히 많은 양의 데이터에서 빠르게 정보를 추출 및 분석하여 가치 있는 정보를 발견하는 기술이다. 다루는 데이터는 어마어마한 대량의 정형(structured, 글자) 또는 비정형(unstructured, 이미지, 동영상) 데이터이다.
(5) 딥 러닝 : 큰 틀에서 사람의 사고방식을 컴퓨터에게 가르치는 기계학습의 한 분야라고 이야기할 수 있다. 어떠한 데이터가 있을 때 이를 컴퓨터가 알아 들을 수 있는 형태로 표현하고 이를 학습에 적용하기 위해 많은 연구가 진행되고 있다.