

2018-국회직-컴퓨터일반-가형-해설

대방고시 전산직/계리직, 하이클래스 군무원 곽후근(gobarian@gmail.com)

해설에 대한 모든 권리는 곽후근(대방고시, 하이클래스)에 있습니다.

1. 데이터베이스에서 트랜잭션에 대한 설명 중 다음의 특성이 의미하는 것으로 옳은 것은?

각 트랜잭션은 다른 트랜잭션과 독립적으로 수행되는 것처럼 보여야 하며, 다른 트랜잭션에 영향을 미치지 않는다.

- ① 고립성(Isolation)
- ② 일관성(Consistency)
- ③ 원자성(Atomicity)
- ④ 지속성(Durability)
- ⑤ 투명성(Transparency)

정답 체크 :

(1) 고립성(격리성) : 수행 중인 트랜잭션이 완료될 때까지 다른 트랜잭션들이 중간 연산 결과에 접근할 수 없음을 의미한다.

오답 체크 :

(2) 일관성 : 트랜잭션이 성공적으로 수행된 후에도 데이터베이스가 일관성 있는 상태를 유지해야 함을 의미한다.

(3) 원자성 : 트랜잭션의 연산들이 모두 정상적으로 실행되거나 하나도 실행되지 않아야 하는 all-or-nothing 방식을 의미한다.

(4) 지속성 : 트랜잭션이 성공적으로 완료된 후 데이터베이스에 반영한 수행 결과는 영구적이어야 함을 의미한다.

(5) 투명성 : DDBMS(분산 DBMS)의 특성으로 트랜잭션이 발생해도 분산 데이터베이스의 무결성과 일관성이 유지됨을 의미한다.

2. 프로세스의 상태 변이에 대한 설명으로 옳지 않은 것은?

① 시간 할당량(time slice)을 사용하는 일반적 우선순위 기반 스케줄링에서 실행(running) 상태 프로세스의 시간 할당량이 모두 소진되었을 때, 우선순위가 높은 다른 준비 상태의 프로세스가 있다면 실행 중이던 프로세스는 커널(kernel)에 의해 스케줄링되기를 기다리는 준비(ready) 상태로 전이된다.

② 실행 상태의 프로세스가 동기식 입출력 요청을 하면, 일반적으로 해당 프로세스는 입출력이 완료될 때까지 CPU를 반납하고 대기(blocked 또는 waiting) 상태로 전이된다.

③ 대기 상태의 프로세스가 요청하였던 입출력이 완료되면, 해당 프로세스는 CPU 연산이 가능해지므로 바로 실행 상태로 전이된다.

④ 다중 처리기 시스템(multi-processor system)에서는 실행 상태의 프로세스가 여러 개 있을 수 있다.

⑤ 대기 상태의 프로세스들은 CPU 할당을 위한 스케줄링에서 제외된다.

정답 체크 :

(3) 입출력이 완료되면 실행 상태가 아닌 준비 상태로 간다.

오답 체크 :

프로세스의 상태 변이를 그림으로 나타내면 다음과 같다.

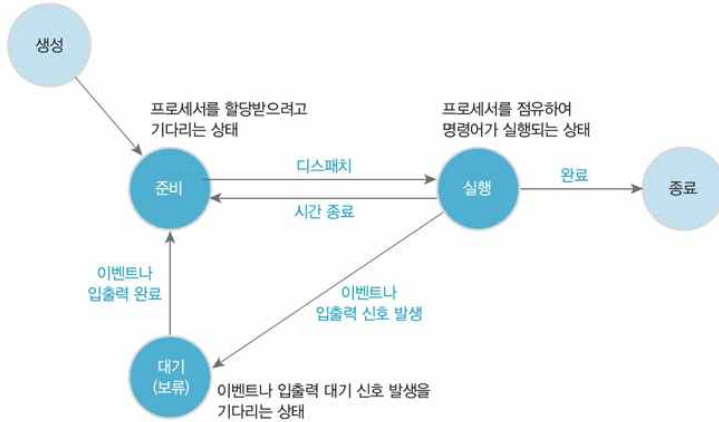


그림 3-5 프로세스의 상태 변화

- (1) 우선순위 스케줄링에서 자신에게 주어진 시간이 끝나면, 준비 상태로 간다.
- (2) 실행 상태에서 동기식 입출력 요청을 하면 대기 상태가 되지만, 비동기식 입출력 요청을 하면 대기 상태가 아닌 실행 상태를 유지한다.
- (4) 다중 처리기는 CPU가 여러 개이기 때문에 여러개의 프로세스를 동시에 실행할 수 있다.
- (5) 대기 상태의 프로세스들은 스케줄링에서 제외되고, 준비 상태의 프로세스들을 중심으로 스케줄링이 수행된다.

3. OSI(Open System Interconnection) 7계층에 대한 설명으로 옳지 않은 것은?

- ① 7계층에서 물리적 전송 매체와 직접적으로 연관이 있는 계층은 물리 계층이다.
- ② 수신 측에서 패킷을 수신하게 되면, 상위 계층에서 하위 계층 순으로 처리된다.
- ③ 전송 계층(transport layer)의 대표적인 프로토콜로는 TCP, UDP 등이 있다.
- ④ 네트워크 계층에서는 라우팅 및 패킷 전송의 기능을 수행한다.
- ⑤ 응용 계층에서는 전자 사서함, 파일 전송 등의 서비스를 제공한다.

정답 체크 :

(2) 수신측에서 패킷을 수신하게 되면, 하위 계층에서 상위 계층 순으로 처리된다(디캡슐화).

오답 체크 :

- (1) 물리 계층 : 데이터 링크층으로 부터 한 단위의 데이터를 받아 통신 링크를 따라 전송될 수 있는 형태로 변환한다. 회선 구성, 데이터 전송 모드, 접속형태, 신호, 부호화, 인터페이스, 전송매체 등을 고려한다.
- (3) 전송 계층 : 메시지 종단(End-to-End) 전달, 포트 주소 지정, 단편화와 재조립, 연결 제어(관리), 흐름 제어, 혼잡 제어 등을 수행한다. 프로토콜에는 TCP, UDP, SCTP 등이 있다.
- (4) 네트워크 계층 : 해당 설명은 표현 계층이고, 네트워크 계층은 패킷 종단(End-to-End) 전달, 논리적인(IP) 주소 지정, 경로 지정(Routing), 주소 변환(ARP) 등을 수행한다.
- (5) 응용 계층 : 네트워크 상의 소프트웨어 사용자에게 사용자 인터페이스 제공한다. 전자우편(X.400), 원격파일 접근과 전송(FTAM), 공유 데이터베이스 관리 및 여러 종류의 분산 정보 서비스(X.500) 제공한다.

4. 다음 정렬 알고리즘의 수행시간을 Big-O 표기법으로 나타냈을 때 최악의 경우의 수행시간이 같은 것들로만 나열된 것은?

선택 정렬(selection sort), 합병 정렬(merge sort),
삽입 정렬(insertion sort), 퀵 정렬(quick sort), 힙 정렬(heap sort)

- ① 합병 정렬, 퀵 정렬, 힙 정렬
- ② 힙 정렬, 선택 정렬, 퀵 정렬
- ③ 합병 정렬, 선택 정렬, 삽입 정렬
- ④ 합병 정렬, 힙 정렬, 삽입 정렬
- ⑤ 선택 정렬, 삽입 정렬, 퀵 정렬

정답 체크 :

(5) 선택 정렬, 삽입 정렬, 퀵 정렬 : 최악의 경우 수행 시간이 $O(n^2)$ 이다.

오답 체크 :

(1), (2), (3), (4) : 합병 정렬과 힙 정렬은 최악의 경우 수행 시간이 $O(n\log n)$ 이다.

5. 관계형 데이터베이스(relational database)에 대한 설명으로 옳은 것을 <보기>에서 모두 고르면?

<보 기>

- ㄱ. 스키마 변환 시 정보의 무손실, 자료 중복의 감소, 관련된 구조 간의 통합의 원칙을 준수하여야 한다.
- ㄴ. 관계대수(relational algebra)의 연산에서 피연산자는 모두 릴레이션이지만 연산결과는 릴레이션이 아니다.
- ㄷ. 릴레이션에 연산을 수행 시 삽입이상(insertion anomaly), 삭제이상(deletion anomaly), 갱신이상(update anomaly)이 발생할 수 있다.
- ㄹ. 튜플을 구성하는 속성 사이에 존재하는 종속관계를 고려하지 않고 하나의 릴레이션으로 표현하여 이상(anomaly)을 해결 할 수 있다.
- ㅁ. 릴레이션이 여러 속성을 표현할 때 이를 작게 분해(decomposition)하는 과정을 정규화(normalization)라고 한다.
- ㅂ. 릴레이션들은 관계대수(relational algebra)로 조작성 가능하다.

- ① ㄱ, ㄴ, ㄷ
- ② ㄴ, ㄷ, ㄹ
- ③ ㄴ, ㄹ, ㅁ
- ④ ㄷ, ㄹ, ㅂ
- ⑤ ㄷ, ㅁ, ㅂ

정답 체크 :

(5)

(ㄷ) : 삽입, 삭제, 갱신 이상이 발생한다.

(ㅁ) : 작게 무손실 분해하는 과정을 정규화라고 한다.

(ㅂ) : 릴레이션은 관계 대수(일반 집합 4개, 순수 관계 4개)로 조작성 가능하다.

오답 체크 :

(1), (2), (3), (4)

(ㄱ) : 관련된 구조 간의 통합이 아닌 분리의 원칙을 준수하여야 한다.

(ㄴ) : 연산결과도 릴레이션이다.

(ㄹ) : 종속 관계를 고려하지 않으면 이상 문제를 해결할 수 없다.

6. 후위표기법(postfix notation)으로 된 다음 식의 전위표기법(prefix notation)으로 옳은 것은?

ABC+D/-AE+BF*/+

- ① +-A/+BCD/+AE*BF
- ② -+A/BC+D/+AE*BF
- ③ +-A/+BCD/+*AE*BF
- ④ +A-/+BCD/+AE*BF
- ⑤ -+/A+BCD/+*AE*BF

정답 체크 :

(1)

후위 표기법을 전위 표기법으로 바꾸는 방법은 다음과 같다.

- 오퍼랜드를 만나면 스택에 push한다.
- 연산자(op)를 만나면 스택에서 2개(s1, s2)의 오퍼랜드를 꺼내고 op s1 s2로 만들어 스택에 push한다. (순서에 주의한다. 스택 상에서 s1이 s2 아래에 있다.)
- 수식의 끝까지 반복한다.

이를 기준으로 주어진 조건을 후위 표기식에서 전위 표기식으로 바꾸면 다음과 같다.

- 오퍼랜드 A, B, C를 스택에 push한다.
- 연산자(+)를 만나면 스택에서 최상위 오퍼랜드 2개(B, C)를 꺼내 +BC로 만들어서 스택에 push한다.
- 이와 동일한 방법으로 수식의 끝까지 하게 되면 마지막 결과가 나오게 된다.

				F			
C	D		E	B	*BF		
B	+BC	/+BCD	A	+AE	+AE	/+AE*BF	
A	A	A	-A/+BCD	-A/+BCD	-A/+BCD	-A/+BCD	+-A/+BCD/+AE*BF

7. 다음 카르노 맵에 해당하는 논리식 F로 옳은 것은?

	A	0	1
BC	00	01	11
10	0	0	0
1	1	0	0

- ① $F = AB + AB'C$
- ② $F = A'B' + A'B'C'$
- ③ $F = AB' + A'BC$
- ④ $F = A'B + AB'C'$
- ⑤ $F = AB + BC$

정답 체크 :

(4)

주어진 카노맵을 간략화하면 다음과 같다.

BC \ A	0	1
00	0	1
01	0	0
11	1	0
10	1	0

$AB'C'$

$A'B$

8. 가상 메모리 시스템에서 메모리 부족 시의 페이지 교체 기법(page replacement algorithm) 들에 대한 설명으로 옳지 않은 것은?

- ① LRU(Least Recently Used) 기법은 메모리에 적재된 페이지 중 가장 오랫동안 참조되지 않았던 페이지를 교체하는 기법이다.
- ② LRU 기법은 실제 그 구현 오버헤드가 커서, 일반적으로 오버헤드를 줄인 여러 유형의 LRU 근사 알고리즘(LRU approximation algorithm)들이 사용되는 것이 보통이다.
- ③ LRU 기법은 물리적 페이지의 개수를 확장했음에도 페이지 폴트 가 늘어나는 경우가 발생할 수도 있는데, 이를 Belady's anomaly 라 한다.
- ④ 이론적으로는 최적의 페이지 교체 기법은 메모리에 적재된 페이지들 중에서 앞으로 가장 오랫동안 참조되지 않을 페이지를 교체 하는 것이다.
- ⑤ FIFO(First In First Out) 기법은 메모리에 적재된 페이지들 중 가장 먼저 메모리에 적재된 페이지를 교체하는 방법이다.

정답 체크 :

(3) Belady's anomaly는 FIFO에서 발생한다.

오답 체크 :

- (1) LRU는 최근에 사용했던 것이 앞으로도 사용할 확률이 높다는 것에 기반한다(시간적 지역성).
- (2) LRU 근사 알고리즘에는 참조비트, 시계-2차적기회, NUR, LFU, MFU 등이 존재한다.
- (4) 최적의 방법으로 구현하기가 어렵다. 왜냐하면 앞으로 가장 오랫동안 참조되지 않을 페이지를 찾는 것은 아주 어렵기 때문이다.
- (5) FIFO는 처음 들어온 페이지를 먼저 교체하므로 시간상으로 오래된 페이지를 먼저 교체하는 것이다.

9. 요즘 집안에서 스마트폰 및 컴퓨터의 인터넷 접속을 위하여 공유 기의 사용이 일반화되어 있다. 공유기에 접속하는 컴퓨터 디바이스에는 사설 IP가 주로 할당되는데 이때 IP주소의 자동 할당을 위해 사용되는 프로토콜로 옳은 것은?

- ① ARP
- ② DHCP
- ③ TCP
- ④ UDP
- ⑤ ICMP

정답 체크 :

(2) DHCP : 호스트가 네트워크에 접속하고자 할때마다 IP를 동적으로 할당한다. 예를 들어, 커피숍에 가면 와이파이에 접속하게 되는데 와이파이에서 DHCP를 이용해서 사용자에게 IP와 임대 기간을 할당한다.

오답 체크 :

(1) ARP : IP 주소(논리 주소)를 MAC 주소(물리 주소)로 바꿔준다. RARP는 MAC 주소를 IP 주소로 바꿔준다.

(3) TCP : 근거리 통신망이나 인트라넷, 인터넷에 연결된 컴퓨터에서 실행되는 프로그램 간에 일련의 옥텟을 안정적으로, 순서대로, 어려없이 교환할 수 있게 한다. 연결 설정을 수행하고, 흐름 제어와 혼잡 제어를 수행한다. TCP는 웹 브라우저들이 월드 와이드 웹에서 서버에 연결할 때 사용되며, 이메일 전송이나 파일 전송에도 사용된다.

(4) UDP : 연결을 설정하지 않고 수신자가 데이터를 받을 준비를 확인하는 단계를 거치지 않고 단방향으로 정보를 전송한다. UDP를 사용하는 애플리케이션에는 도메인 이름 서비스(DNS), IPTV, 음성 인터넷 프로토콜(VoIP), TFTP, IP 터널, 그리고 많은 온라인 게임 등이 있다.

(5) 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.

10. 다음의 C 프로그램 실행 결과로 출력되는 a, b, c 값으로 옳은 것은?

```
#include <stdio.h>
int foo(int a, int *b)
{
    int c;
    *b = a + 1;
    c = a - 1;
    return c;
}
void main()
{
    int a = 5;
    int b = 3;
    int c = 0;
    b = foo(a, &c);
    c = foo(b, &a);
    printf("a=%d b=%d c=%d\n", a, b, c);
}
```

- ① 5, 4, 3
- ② 4, 3, 2
- ③ 6, 3, 4
- ④ 4, 3, 5

⑤ 5, 3, 0

정답 체크 :

(1)

a = 5, b = 3, c = 0; // 초기 조건이다.

b = foo(a, &c); // a의 값은 변하지 않고, b와 c의 값은 변한다.

c = 6; return 4; // foo를 실행하면 b는 4가 되고, c는 6이 된다.

c = foo(b, &a); // b의 값은 변하지 않고, a와 c의 값은 변한다. 지문에서 b가 4인 답은 (1) 번밖에 없기 때문에 더 이상 코드를 분석할 필요가 없다.

a = 5; return 3; // foo를 실행하면 a는 5가 되고, c는 3이 된다.

a = 5, b = 4, c = 3 // 최종 결과이다.

Tip! : 해당 문제는 자주 출제되는 유형으로 call-by-value와 call-by-reference를 묻는 질문이다. 2가지의 특성을 잘 이용하면 중간에 답이 나올 수 있으므로 문제를 다 풀지 말고 시간을 절약하기 바란다.

11. 데이터베이스에서 사용하는 뷰(view)에 대한 설명으로 옳은 것을 <보기>에서 모두 고르면?

- | |
|---|
| <p><보 기></p> <ul style="list-style-type: none"> ㄱ. 뷰의 정의는 변경할 수 없다. ㄴ. 뷰는 삽입, 갱신, 삭제 연산에 제약이 있다. ㄷ. 뷰 위에 다른 뷰를 정의할 수 없다. ㄹ. 뷰가 정의된 테이블이 삭제되더라도 뷰는 남는다. ㅁ. 뷰는 물리적으로 구현되는 테이블이다. |
|---|

① ㄱ, ㄴ

② ㄱ, ㅁ

③ ㄴ, ㄷ

④ ㄷ, ㄹ

⑤ ㄹ, ㅁ

정답 체크 :

(1)

(ㄱ) : 일반적으로 뷰의 정의는 변경할 수 없지만, ALTER VIEW와 같은 명령어를 제공하는 데이터베이스도 존재한다. (공무원 시험의 특성 상 가장 근접한 답의 측면에서 바라보면 맞는 답이다.)

(ㄴ) : 뷰에 대한 삽입·수정·삭제 연산은 제한적으로 수행된다. 변경 가능한 뷰와 변경 불가능한 뷰가 존재한다. 예를 들어 기본 테이블의 기본키를 구성하는 속성이 포함되어 있지 않은 뷰나 기본 테이블에 있던 내용이 아닌 집계 함수로 새로 계산된 내용을 포함하는 뷰 등은 변경 불가능한 뷰이다.

오답 체크 :

(2), (3), (4), (5)

(ㄷ) : 뷰 위에 다른 뷰를 정의할 수 있다.

(ㄹ) : 뷰가 정의된 테이블이 삭제되면 뷰도 삭제된다. 물리적인 테이블이 삭제되었는데 논리적인 테이블이 남아 있는 것은 이치상 맞지 않는다.

(㉑) : 뷰는 논리적으로 구현되는 테이블이다.

12. 프로세스(process)나 스레드(thread)들이 공유 자원에 하나 이상의 수정(write 또는 modify) 연산을 포함하는 동시 접근을 할 때 그 접근 부분들을 임계 구역(critical section)이라 한다. 이를 보호하기 위한 병행 프로세스 동기화 기법으로 옳은 것은?

- ① 인터럽트(interrupt)
- ② 선점 스케줄링(preemptive scheduling)
- ③ 문맥교환(context switching)
- ④ 상호배제(mutual exclusion)
- ⑤ 교착상태(deadlock)

정답 체크 :

(4) 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.

오답 체크 :

(1) 인터럽트 : 프로그램을 실행하고 있을 때, 입출력 하드웨어 등의 장치나 또는 예외상황이 발생하여 처리가 필요할 경우에 마이크로프로세서(CPU)에게 알려 처리할 수 있도록 하는 것을 말한다.

(2) 선점 스케줄링 : 시분할 시스템에서 타임 슬라이스가 소진되었거나, 인터럽트나 시스템 호출 종료 시에 더 높은 우선 순위 프로세스가 발생 되었음을 알았을 때, 현 실행 프로세스로부터 강제로 CPU를 회수하는 것을 말한다.

(3) 문맥교환 : 실행 중인 프로세스의 제어를 다른 프로세스에 넘겨 실행 상태가 되도록 하는 것이다. 프로세스 문맥 교환이 일어나면 프로세서의 레지스터에 있던 내용 저장한다.

(5) 교착상태 : 다중 프로그래밍 시스템에서, 프로세스가 결코 일어나지 않을 사건(event)을 기다리는 상태이다. 프로세스가 교착 상태에 빠지면, 작업 정지되어 명령 진행 불가하다.

13. 배열 `int array[10][200]`를 행우선순서(row major order)로 저장하는 경우의 원소 `array[7][12]`의 시작주소는 몇 번지인가? (단, 배열 `array`의 시작주소는 `10840h`로, `int`의 크기는 4바이트로 가정한다. 배열 첨자는 0부터 시작하며 숫자에 붙은 `h`는 16진수 표기를 의미한다.)

- ① 10804h
- ② 11E50h
- ③ 16488h
- ④ 108BFh
- ⑤ 10A3Ch

정답 체크 :

(2)

`int A[row][col];` // 2차원 배열이 선언되었다고 가정한다.

`A[0][0];` // 시작 주소는 `A[0][0]` 혹은 `A`이다.

요소크기 = `int` = 4 bytes; // 요소크기는 `int`이기 때문에 4바이트이다. 만약, 요소가 `int`가 아니라 `char`(1바이트) 혹은 `double`(8바이트)이면 요소크기에 주의해야 한다.

$A[i][j]$ 의 주소 = 시작주소 + $(i * \text{col} + j) * \text{요소크기}$; // 행 우선 순서로 저장할 때는 주어진 수식으로 개별 요소의 주소를 계산할 수 있다. (행을 중심으로 배열의 요소를 배치하므로 $i * \text{col}$ 로 계산된다.)

$A[i][j]$ 의 주소 = 시작주소 + $(j * \text{row} + i) * \text{요소크기}$; // 열 우선 순서로 저장할 때는 주어진 수식으로 개별 요소의 주소를 계산할 수 있다. (열을 중심으로 배열의 요소를 배치하므로 $j * \text{row}$ 로 계산된다. 이는 행 우선 순서와 역의 관계를 가진다.)

주어진 조건으로 문제를 풀면 다음과 같다.

`int array[10][200];` // 주어진 2차원 배열이다.

시작주소 = 10840h // 시작 주소이다.

요소크기 = int = 4바이트 // 요소크기는 int이다.

`array[7][12]`의 주소 = 10840h + $(7 * 200 + 12) * 4 = 11E50h$; // 행 우선 순위로 계산한 주소값이다.

Tip! : 행 우선 순서와 열 우선 순서는 역의 관계를 가지므로 행 우선 순서 하나만 이해하고 외우면 된다.

14. 다음은 마트에서 판매 기록을 저장한 트랜잭션(transaction) 데이터이다. 규칙 “기저귀” -> “맥주”의 신뢰도(confidence) 값으로 옳은 것은?

식별자	품목
1	맥주, 땅콩, 기저귀
2	맥주, 커피, 기저귀
3	맥주, 기저귀, 계란
4	땅콩, 계란, 우유
5	땅콩, 커피, 기저귀, 우유

- ① 0.33
- ② 0.5
- ③ 0.75
- ④ 0.8
- ⑤ 1.00

정답 체크 :

(3)

정보추출은 고객관련 정보를 토대로 미래의 구매형태를 예측하거나 변수 간 인과관계를 분석하는 마케팅 기법이다. 정보추출의 예에는 연관 규칙이 있다. 연관 규칙이란 하나의 거래나 사건에 포함되어있는 품목들의 상호 연관성을 발견하는 것이다. 연관성은 어떤 item 집합의 존재가 다른 item 집합의 존재를 암시하는 것을 의미하여 $A \rightarrow B$ (만일 A가 일어나면 B가 일어남)와 같이 표시한다. 함께 구매하는 상품의 조합이나 서비스 패턴 발견할 수 있다. 미국 월마트 기저귀와 맥주 판매를 보면 기저귀가 많이 판매되면 맥주도 많이 판매가 되는 패턴을 알 수 있다(아마도 어린 아이들을 키우는 부모들은 육아의 지침을 맥주로 달래는 것이다. 이것은 아이를 양육해보면 바로 알 수 있다^^). 연관 규칙 분석 방법은 지지도(support)와 신뢰도(confidence)가 있다.

지지도는 전체 거래 중 X(기저귀)와 Y(맥주)가 동시에 포함하는 거래가 어느 정도인가를 나타낸다. 여기서 N은 전체 트랜잭션의 개수를 나타내고, 연관 규칙 $X \rightarrow Y$ 는 지지도 S를 갖는다.

$$S = \frac{|X \cap Y|}{N}$$

신뢰도는 X를 포함하는 거래 중에서 Y가 포함된 거래는 어느 정도인가를 나타낸다. 연관 규칙 $X \rightarrow Y$ 는 신뢰도 C를 갖는다.

$$S = \frac{|X \cap Y|}{|X|}$$

위의 수식을 통해 계산하면 지지도는 $0.6(=3/5)$ 이 되고, 신뢰도는 $0.75(=3/4)$ 가 된다.

15. 사용자가 WWW(World Wide Web)을 사용하여 임의의 사이트에 접속하는 과정의 설명으로 옳지 않은 것은?

- ① 사용자의 웹브라우저에서 DNS 서버로 DNS 질의 메시지를 보낸다.
- ② DNS 간의 질의는 재귀형(recursive) 또는 반복형(iterative)으로 연계된다.
- ③ 사용자의 웹브라우저는 DNS 서버로부터 사이트의 웹 서버 MAC 주소를 획득한다.
- ④ HTTP 요청 메시지를 보내기 전에 TCP 3-방향 핸드셰이크를 수행한다.
- ⑤ 웹 서버가 보낸 HTTP 응답 메시지의 바디로부터 HTML을 추출하여 웹 페이지를 보여준다.

정답 체크 :

(3) 사용자의 웹브라우저는 DNS 서버로부터 사이트의 웹 서버 IP 주소를 획득한다.

오답 체크 :

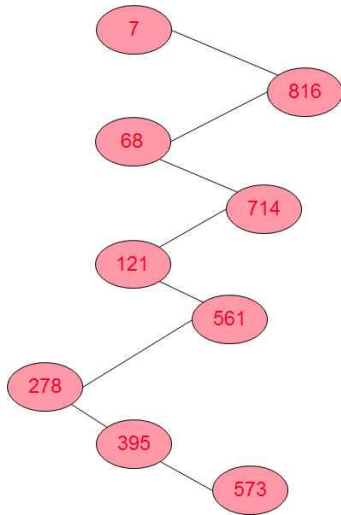
- (1) 사용자의 웹브라우저는 사이트의 IP 주소를 얻기 위해 DNS 서버로 DNS query를 보낸다.
- (2) DNS query는 recursive, iterative, non-recursive가 있다. DNS는 계층적 구조를 가지면 5개의 계층이 있다고 가정하면 recursive는 첫 번째 DNS가 두 번째 DNS에게 요청하고, 두 번째 DNS가 세 번째 DNS에게 요청하는 방식이다. iterative는 첫 번째 DNS가 두 번째 DNS에게 요청하고, 다시 첫 번째 DNS가 두 번째 DNS에게 요청하는 방식이다. 마지막으로 non-recursive는 첫 번째 DNS가 다른 DNS 서버에게 요청해서 부분적인 결과만을 얻는다. 이로 인해 나중에 실제로 연결을 맺으려고 할 때 DNS로 인한 초기 시간을 줄일 수 있다.
- (4) HTTP는 TCP 위에서 동작하기 때문에 TCP 3-way handshake를 수행한다. 일반적인 방법은 아니지만 HTTP를 UDP 위에서 동작시킬 수도 있다.
- (5) 결국 HTTP란 HTML을 주고 받기 위한 프로토콜인 것이다.

16. 1과 1000 사이의 정수로 구성된 이진탐색트리(binary search tree)에서 숫자 573을 탐색하는 경우 다음 중 비교경로로 옳지 않은 것은?

- ① 2, 173, 241, 856, 301, 489, 710, 516, 573
- ② 7, 816, 68, 714, 121, 561, 278, 395, 573
- ③ 981, 825, 693, 38, 137, 608, 224, 461, 573
- ④ 926, 139, 884, 278, 734, 319, 662, 481, 573
- ⑤ 14, 970, 831, 765, 111, 249, 318, 473, 573

정답 체크 :

(2) 해당 조건을 그림으로 그리면 다음과 같다. 이진탐색트리의 조건은 자신의 노드를 기준으로 왼쪽 자식은 자신보다 같거나 작고, 오른쪽 자식은 자신보다 크거나 같다는 전제 조건에서 출발한다. 816에서 왼쪽 자식(86)을 비교했기 때문에 이후의 비교 숫자는 816보다 적어야 한다. 마찬가지로 조건으로 714보다 적어야 하고, 561보다 적어야 한다. 그런데 마지막 비교 숫자가 573으로 이는 561보다 크기 때문에 잘못된 비교 경로가 된다. 즉, 561에서 왼쪽의 비교 경로로 온 이상 이진탐색트리의 조건에 의해 561보다 큰 숫자가 나올 수 없다.



오답 체크 :

(1), (3), (4), (5) : 그림을 그려서 확인해보면 모두 다 이진탐색트리에서 정상적인 비교 경로를 가짐을 알 수 있다.

17. 유니버설 게이트 집합(universal gate set) 은 그 구성 원소만으로 어떤 형태의 디지털 시스템도 구현할 수 있는 기능적으로 완전한 게이트들의 집합이다. 다음 중 유니버설 게이트 집합으로 옳지 않은 것은?

- ① { NAND }
- ② { OR, DECODER }
- ③ { MULTIPLEXER }
- ④ { OR, NOT }
- ⑤ { XOR }

정답 체크 :

(5) XOR : 유니버설 게이트는 AND, OR, NOT만을 만들 수 있다. 물론, XOR도 AND, OR, NOT의 조합으로 만들 수 있기는 하지만 해당 문제에서는 간접적으로 만들어지는 것이 아닌 직접적으로 만들어지는 것을 대상으로 한다. 그러므로 XOR은 가장 틀린 답이 된다.

오답 체크 :

- (1) NAND : 유니버설 게이트이다.
- (2) OR, DECODER : OR는 유니버설 게이트 집합으로 만들 수 있고, 디코더는 AND와 NOT으로 이루어져 있으므로 이들은 유니버설 게이트 집합으로 만들 수 있다.
- (3) MULTIPLEXOR : 멀티플렉서는 AND, OR, NOT으로 구성되고 이들은 유니버설 게이트 집합으로 만들 수 있다.

(4) OR, NOT : OR, NOT은 유니버설 게이트 집합으로 만들 수 있다.

18. 다음 C 프로그램의 실행 결과로 옳은 것은?

```
#include <stdio.h>
int recursion(int n)
{
    if (n < 5) return 1;
    else if (n % 5 == 1) return n + recursion(n - 1);
    else recursion(n - 1);
}
int main()
{
    int n = recursion(16);
    printf("%d", n);
    return 0;
}
```

- ① 34
- ② 33
- ③ 31
- ④ 29
- ⑤ 28

정답 체크 :

(1)

recursion(16); // recursion(16)을 호출한다.

return 16 + recursion(15); // if (n%5 == 1)을 만족하기 때문에 16을 리턴하고 recursion(15)를 호출한다.

recursion(15); -> 14 -> 13 -> 12 -> 11 // if (n%5 == 1)을 만족하지 않기 때문에 recursion(14), recursion(13), recursion(12), recursion(11)을 차례대로 호출한다.

return 11 + recursion(10); // if (n%5 == 1)을 만족하기 때문에 11을 리턴하고 recursion(10)을 호출한다.

recursion(10); -> 9 -> 8 -> 7 -> 6 // if (n%5 == 1)을 만족하지 않기 때문에 recursion(9), recursion(8), recursion(7), recursion(6)을 차례대로 호출한다.

return 6 + recursion(5); // if (n%5 == 1)을 만족하기 때문에 6을 리턴하고 recursion(5)를 호출한다.

recursion(5); -> 4 // if (n%5 == 1)을 만족하지 않기 때문에 recursion(4)를 호출한다.

return 1; // if (n < 5)를 만족하기 때문에 1을 리턴한다.

n = 16 + 11 + 6 + 1 = 34; // 리턴된 값들을 더하면 34가 된다.

Tip! : 순환 호출(재귀 함수)은 자주 출제되는 문제이다. 일단, 순환 호출의 개념과 종료 조건 파악이 중요하며 규칙성을 찾는 것이 매우 중요하다. 만약, 규칙성을 찾을 수 있다면 더 이상 코드를 보지 않고 문제를 풀 수 있다.

19. 인터넷에서 주로 사용하는 TCP와 UDP에 대한 설명으로 옳지 않은 것은?

- ① TCP에는 네트워크 상 트래픽이 폭주하는 현상을 방지하는 혼잡 제어(congestion control) 기능이 포함되어 있다.
- ② UDP는 비신뢰적이고 비연결형인 서비스를 제공한다.
- ③ TCP는 패킷 손실이 발생했을 때, 패킷 재전송 기능을 수행한다.
- ④ UDP 패킷 헤더에는 출발지 포트 번호, 목적지 포트 번호, 길이, 체크섬 등의 정보를 담고 있다.
- ⑤ TCP는 신뢰적인 데이터 전달은 보장하지만, 수신 프로세스에게 데이터가 올바른 순서로 전달되는 것은 보장하지 못한다.

정답 체크 :

(5) TCP는 신뢰적인 데이터 전달을 보장하고, 수신 프로세스에게 데이터가 올바른 순서로 전달되는 것을 보장한다.

오답 체크 :

- (1) TCP는 혼잡 제어(congestion control)와 흐름 제어(flow control) 기능이 포함되어 있다.
- (2) UDP는 비신뢰적이고 비연결형이다. 즉, 신뢰적인 데이터 전달을 보장하지 않는다.
- (3) TCP는 패킷 재전송(retransmission) 기능을 가진다.
- (4) UDP는 헤더에 출발지 포트 번호, 목적지 포트 번호, 길이, 체크섬의 정보를 가진다.

20. 다음 중 회선 교환 방식(circuit switching)의 특성으로 옳지 않은 것은?

- ① 데이터를 주고받기 전에 종단 간 연결(end to end connection) 과정이 필요하다.
- ② 연결이 수립되면, 해당 회선을 독점적으로 사용한다.
- ③ 회선 교환 방식의 대표적인 예로는 전화망이 있다.
- ④ 패킷 교환 방식(packet switching)에 비해 더 많은 동시 접속자를 수용할 수 있다.
- ⑤ 패킷 교환 방식에 비해 안정적인 전송속도를 보장할 수 있다.

정답 체크 :

(4) 회선 교환 방식은 1:1 연결 방식이고, 패킷 교환 방식은 N:N 연결 방식이다. 그러므로 패킷 교환 방식이 더 많은 동시 접속자를 수용할 수 있다.

오답 체크 :

- (1) 연결 설정 및 해제 과정이 필요하다.
- (2) 발신자와 수신자 간에 독립적이며 동시에 폐쇄적인 통신 연결로 구성되어 있다.
- (3) 전화(PSTN) 또는 컴퓨터와 같은 두 장치를 직접 물리적으로 연결한다.
- (5) 일단 설정된 통신은 안정적이다. 다른 요인에 의해 통신이 방해 받지 않는다.