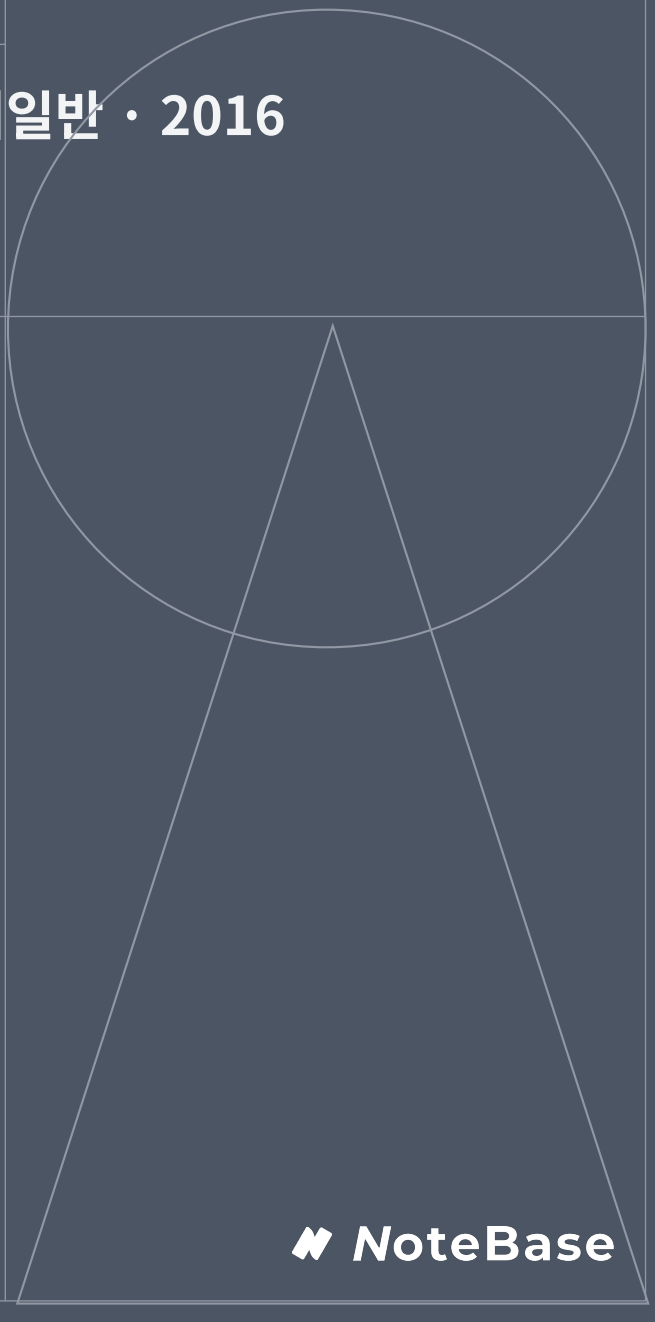
기출OX 해설집

공무원/국가직/9급 · 컴퓨터일반 · 2016



NoteBase

기출OX 해설집

공무원/국가직/9급 · 컴퓨터일반 · 2016

공무원/국가직/9급 · 컴퓨터일반 · 2016

문 1. 데이터베이스 관리 시스템(database management system)을 구축함으로써 생기는 이점만을 모두 고른 것은?

ㄱ. 응용 소프트웨어가 데이터베이스에 관한 세부 사항에 자세히 관련할 필요가 없어져서 응용 소프트웨어 설계가 단순화될 수 있다.
ㄴ. 데이터베이스에 대한 접근 제어가 용이해진다.
ㄷ. 데이터 독립성을 제거할 수 있다.
ㄹ. 응용 소프트웨어가 데이터베이스를 직접 조작하게 된다.

① ㄱ, ㄴ
② ㄱ, ㄷ
③ ㄴ, ㄹ
④ ㄷ, ㄹ

정답 1번

간단한 해설

DBMS는 데이터의 저장·처리 세부 사항을 대신 관리하므로 응용 소프트웨어 설계를 단순화할 수 있고, 사용자별 권한을 관리하여 **접근 제어**를 쉽게 한다. 응용 소프트웨어는 **DBMS를 통해** 데이터에 접근하며, 그 과정에서 데이터 독립성을 확보할 수 있다.

NoteBase

보기별 해설

ㄱ. O: DBMS가 데이터의 저장 위치와 저장 방식 등 세부 사항을 관리한다. 응용 소프트웨어는 필요한 데이터를 요청하면 되므로, 저장 파일을 읽고 쓰는 세부 기능을 직접 구현하는 부담이 줄어 **설계가 단순화될 수 있다**.

ㄴ. O: DBMS는 사용자별로 데이터의 조회·삽입·수정·삭제 권한을 관리하고 접근 요청에 해당 권한이 있는지 검사한다. 예를 들어 조회 권한만 부여한 사용자의 수정 요청을 거부할 수 있으므로 **접근 제어를 일관되게 적용**하기 쉽다.

ㄷ. X: DBMS는 데이터 독립성을 제거하는 것이 아니라 **데이터 독립성을 제공**한다. 데이터 독립성이란 데이터 구조나 저장 방식의 변화가 응용 프로그램에 미치는 영향을 줄이는 성질이다. 이를 통해 데이터 저장 방식을 바꿀 때 응용 프로그램까지 함께 수정해야 하는 부담을 줄일 수 있다.

ㄹ. X: 응용 소프트웨어는 데이터베이스의 저장 내용을 직접 조작하는 것이 아니라 **DBMS에 요청하여 데이터를 조작**한다. DBMS가 그 요청을 처리하고 실제 데이터의 조회나 변경을 수행하므로, 응용 소프트웨어와 저장된 데이터 사이의 중개 역할을 한다.

DBMS 구축의 이점에 해당하는 보기는 ㄱ, ㄴ이므로, 이를 묶은 선택지는 ①이다.

NoteBase

문 2. 다음은 PC(Personal Computer)의 전원을 켰을 때 일어나는 과정들을 순서대로 나열한 것이다. ㉠~㉢이 바르게 짝지어진 것은?

○ (㉠)에 저장된 바이오스(BIOS)가 실행되어 컴퓨터에 장착된 하드웨어 장치들의 상태를 점검한다.
○ (㉡)에 저장되어 있는 운영체제가 (㉢)로/으로 로드(load)된다.
○ 운영체제의 실행이 시작된다.

① ㉠: 보조기억장치, ㉡: ROM, ㉢: 주기억장치
② ㉠: 보조기억장치, ㉡: 주기억장치, ㉢: ROM
③ ㉠: ROM, ㉡: 보조기억장치, ㉢: 주기억장치
④ ㉠: ROM, ㉡: 주기억장치, ㉢: 보조기억장치

정답 3번

간단한 해설

전원을 켜자마자 실행되는 BIOS는 전원이 꺼져도 내용이 유지되는 **ROM**에 저장된다. 운영체제는 **보조기억장치에서 주기억장치로 로드**된 뒤 실행된다.

빈칸별 해설

㉠ ROM: BIOS는 컴퓨터의 기본적인 하드웨어 점검과 초기화를 담당하는 프로그램이다. 전원을 켰을 때부터 사용할 수 있어야 하므로, **전원이 꺼져도 내용이 유지되는** ROM에 저장된다. BIOS가 하드웨어 상태를 점검하는 과정을 POST라고 한다.

㉡ 보조기억장치: 운영체제는 HDD나 SSD와 같은 보조기억장치에 저장되어 있다. 보조기억장치는 전원을 꺼도 내용을 보존하므로 **운영체제를 지속적으로 저장**하는 데 사용된다.

㉢ 주기억장치: 운영체제를 실행하려면 필요한 코드와 데이터를 보조기억장치에서 **주기억장치인 RAM으로 옮겨야** 한다. 이처럼 실행에 필요한 내용을 메모리에 올리는 것이 로드이다. RAM은 실행 중인 프로그램과 데이터를 담는 공간이며, 일반적으로 전원이 꺼지면 내용이 사라진다. 따라서 운영체제의 저장 장소인 ㉡과 실행을 위해 로드되는 장소인 ㉢을 구별해야 한다.

㉠ ROM, ㉡ 보조기억장치, ㉢ 주기억장치의 조합은 ③과 일치한다.

NoteBase

문 3. 나머지 셋과 다른 부울 함수를 표현하는 것은?

① F = A + A′B

② F = A(A + B)

③ F = AB′ + A

④ F = (A + B)(A + B′)

정답 1번

간단한 해설

①은 **OR 함수** $F = A + B$로 간소화되어 A가 0이어도 B가 1이면 출력이 1이다. 나머지 식은 $F = A$이므로 이 입력에서 출력이 0이다.

선지별 해설

부울 변수 A와 B는 각각 0 또는 1의 값을 갖는다. '+'는 OR, 곱은 AND, '′'는 NOT 연산을 뜻한다. 각 식을 간소화한 뒤 같은 입력에서 출력이 같은지 비교한다.

① 분배법칙을 적용하면 $A + A'B = (A + A')(A + B)$이다. 서로 보수인 두 값의 OR는 1이므로 $A + A' = 1$이고, 따라서 $F = A + B$이다. 즉, **A와 B 중 하나라도 1이면** 출력이 1이다.

② 흡수법칙에 따라 $F = A(A + B) = A$이다. A가 0이면 바깥의 A 때문에 출력이 0이고, A가 1이면 괄호 안도 1이므로 출력이 1이다. 따라서 출력은 **B와 무관하게 A와 같다**.

③ A를 묶으면 $F = AB' + A = A(B' + 1)$이다. 어떤 값과 1을 OR 연산해도 1이므로 $B' + 1 = 1$이고, $F = A$이다. **OR로 연결된 A가 AB′ 항을 흡수**하므로 B의 영향이 사라진다.

④ 분배법칙으로 $(A + B)(A + B') = A + BB'$이다. 서로 보수인 두 값의 AND는 0이므로 $BB' = 0$이고, $F = A + 0 = A$이다. 두 괄호에 각각 들어 있는 **B와 B′의 곱이 0**이 되는 것이 핵심이다.

②·③·④는 모두 $F = A$이고, ①만 $F = A + B$이다. 예를 들어 A=0, B=1을 넣으면 ①의 출력은 1, 나머지 셋의 출력은 0이므로 ①이 다른 함수를 나타낸다.

NoteBase

문 4. CMMI(Capability Maturity Model Integration)의 성숙도 모델에서 표준화된 프로젝트 프로세스가 존재하나 프로젝트 목표 및 활동이 정량적으로 측정되지 못하는 단계는?

① 관리(managed) 단계

② 정의(defined) 단계

③ 초기(initial) 단계

④ 최적화(optimizing) 단계

정답 2번

간단한 해설

정의(defined) 단계는 조직의 표준 프로세스를 프로젝트에 적용하는 단계이다. 표준화는 이루어졌지만 통계적 기법을 이용한 **정량적 관리에는 아직 이르지 않은** 상태라는 점이 핵심이다.

공통 해설

CMMI 성숙도 3단계인 정의(defined) 단계에서는 **조직 차원의 표준 프로세스**를 마련하고, 각 프로젝트가 이를 특성에 맞게 조정하여 사용한다. 즉, 프로젝트마다 개별적으로 관리하는 수준을 넘어 조직 전체에서 일관된 절차와 기준을 적용한다. 발문의 '표준화된 프로젝트 프로세스가 존재'한다는 조건이 이 단계에 해당한다.

발문의 '정량적으로 측정되지 못한다'는 표현은 수치 자료를 전혀 수집하지 않는다는 뜻이 아니라, 프로세스 성과를 통계적으로 분석하고 예측하는 **정량적 관리 수준에 도달하지 못했다**는 뜻으로 이해한다. 이는 성숙도 4단계인 정량적 관리(quantitatively managed) 단계와 구별되는 기준이다. 초기 단계나 프로젝트별 관리 단계는 조직 표준화 수준에 못 미치고, 최적화 단계는 정량적 관리에 기초하여 지속적으로 개선하는 더 높은 수준이므로, 두 조건을 함께 충족하는 단계는 정의 단계이다.

NoteBase

문 5. 다음에서 설명하는 이미지 파일 형식(format)으로 옳은 것은?

◯ 컴퓨서브사에서 이미지 파일 전송 시간을 줄이기 위해 개발한 이미지 파일 압축 형식이다.
◯ RLE(Run Length Encoding) 방식을 응용한 압축 방법을 사용한다.
◯ 사용 가능한 색이 256색으로 제한된다.

① JPEG
② MPEG
③ TIFF
④ GIF

정답 4번

간단한 해설

컴퓨서브사가 개발했고 색상표에 **최대 256색**을 사용하는 이미지 파일 형식은 GIF이다. 다만 GIF의 실제 압축 알고리즘은 LZW이므로 지문의 RLE 설명과 구분해야 한다.

공통 해설

GIF(Graphics Interchange Format)는 컴퓨서브사가 이미지의 저장과 전송을 위해 개발한 파일 형식이다. 이미지의 각 픽셀에 색상값 자체를 기록하는 대신 색상표의 항목을 가리키는 번호를 기록하며, 한 이미지에 사용하는 **색상표의 색 수는 최대 256개**이다. 따라서 '컴퓨서브사에서 개발'과 '256색으로 제한'이라는 특성이 GIF에 대응한다.

압축 알고리즘은 **LZW를 사용하는 무손실 압축**이다. LZW는 데이터에서 나타나는 패턴을 사전에 등록하고 해당 패턴을 코드로 나타낸다. 반면 RLE는 같은 값이 연속되는 구간을 그 값과 반복 횟수로 표현한다. 따라서 지문의 'RLE 방식을 응용한 압축 방법'은 GIF의 압축 알고리즘을 정확하게 설명한 표현이 아니며, GIF의 압축 방식은 LZW로 구분해야 한다.

NoteBase

문 6. 소프트웨어 테스트에 대한 설명으로 옳지 않은 것은?

① 단위(unit) 테스트는 개별적인 모듈에 대한 테스트이며 테스트 드라이버(driver)와 테스트 스텁(stub)을 사용할 수 있다.

② 통합(integration) 테스트는 모듈을 통합하는 방식에 따라 빅뱅(big-bang) 기법, 하향식(top-down) 기법, 상향식(bottom-up) 기법을 사용한다.

③ 시스템(system) 테스트는 모듈들이 통합된 후 넓이 우선 방식 또는 깊이 우선 방식을 사용하여 테스트한다.

④ 인수(acceptance) 테스트는 인수 전에 사용자의 요구 사항이 만족되었는지 테스트한다.

정답 3번

간단한 해설

③의 '넓이 우선 방식 또는 깊이 우선 방식'은 **하향식 통합 테스트**에서 모듈을 통합하는 순서에 관한 설명이다. 시스템 테스트는 **완성된 시스템 전체**가 명세된 기능과 성능 등의 요구 사항을 충족하는지 확인한다.

선지별 해설

① O: 단위 테스트는 개별 모듈이 의도한 대로 동작하는지 확인한다. 다른 모듈이 아직 준비되지 않았다면 **드라이버는 테스트 대상 모듈을 호출**하고, **스텁은 테스트 대상이 호출하는 모듈을 대신**한다. 이를 통해 주변 모듈이 완성되지 않아도 해당 모듈을 시험할 수 있다.

② O: 통합 테스트는 모듈을 결합하면서 모듈 사이의 인터페이스와 상호작용에 오류가 있는지 확인한다. 빅뱅 기법은 모듈들을 한꺼번에 결합하여 시험한다. 하향식 기법은 **상위 모듈부터 아래로**, 상향식 기법은 **하위 모듈부터 위로** 점진적으로 통합한다.

③ X: 넓이 우선과 깊이 우선은 **하향식 통합 테스트에서 모듈을 통합하는 순서**이다. 넓이 우선은 같은 계층의 모듈들을 먼저 통합하고, 깊이 우선은 한 갈래의 모듈들을 아래 계층까지 먼저 통합한다. 시스템 테스트는 통합을 마친 시스템 전체를 대상으로 기능, 성능, 보안 등 명세된 요구 사항의 충족 여부를 확인하는 단계이므로, 선지의 통합 순서 설명과 구별된다.

④ O: 인수 테스트는 시스템을 인수하기 전에 **사용자의 요구 사항과 인수 기준을 충족하는지** 확인한다. 사용자나 고객의 관점에서 실제 업무에 사용할 수 있는지를 판단하여 인수 여부를 결정하는 근거로 삼는다.

NoteBase

문 7. 다음 C 프로그램의 출력 값은?

```
#include <stdio.h>

int func(int n);
int main(void){
    int num;

    printf("%d\n", func(5));
    return 0;
}

int func(int n){
    if (n < 2)
        return n;
    else {
        int i, tmp, current=1, last=0;
        for(i=2; i<=n; i++){
            tmp = current;
            current += last;
            last = tmp;
        }
        return current;
    }
}
```

① 5

② 6

③ 8

④ 9

정답 1번

간단한 해설

current는 이전 두 값을 더한 값으로, last는 **갱신 전 current**로 바뀐다. i = 2부터 5까지 네 번 반복한 뒤 current는 **5**가 된다.

NoteBase

계산 과정

초기 상태: main은 func(5)를 호출하므로 n은 5이다. n < 2가 거짓이어서 else 부분을 실행하며, current = 1, last = 0으로 시작한다.

갱신 규칙: tmp = current는 변경 전 current를 보관한다. current += last는 두 값의 합을 current에 저장하고, last = tmp는 **변경 전 current를 last에 저장**한다. 따라서 매 반복에서 두 변수는 다음 피보나치 수를 구할 수 있는 연속된 두 값으로 갱신된다.

i = 2: tmp에 1을 저장하고, current는 1 + 0 = 1, last는 1이 된다. 반복 후 (current, last)는 (1, 1)이다.

i = 3: tmp에 1을 저장하고, current는 1 + 1 = 2, last는 1이 된다. 반복 후 (current, last)는 (2, 1)이다.

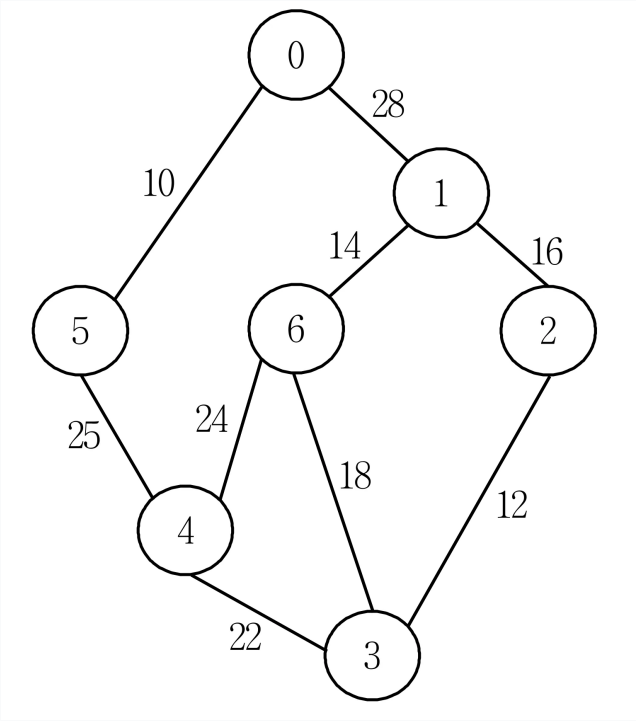
i = 4: tmp에 2를 저장하고, current는 2 + 1 = 3, last는 2가 된다. 반복 후 (current, last)는 (3, 2)이다.

i = 5: 반복 조건이 **i <= n**이므로 i가 5일 때도 실행한다. tmp에 3을 저장하고, current는 3 + 2 = 5, last는 3이 된다. 반복 후 (current, last)는 (5, 3)이다.

반환과 출력: i가 6이 되면 반복 조건이 거짓이 되어 종료한다. return current가 5를 반환하고, printf의 %d가 이를 정수로 출력하므로 출력 값은 5이다. 이는 ①의 값과 일치한다.

NoteBase

문 8. 프림(Prim) 알고리즘을 이용하여 최소 비용 신장 트리를 구하고자 한다. 다음 그림의 노드 0에서 출발할 경우 가장 마지막에 선택되는 간선으로 옳은 것은? (단, 간선 옆의 수는 간선의 비용을 나타낸다)



① (1, 2)

② (1, 6)

③ (4, 5)

④ (4, 6)

정답 2번

간단한 해설

프림 알고리즘은 **현재 트리와 바깥 노드를 잇는 간선 중 비용이 가장 작은 것**을 고른다. 마지막에는 노드 6만 남으므로, 노드 6으로 연결되는 비용 14·18·24의 간선 중 **(1, 6)**을 선택한다.

NoteBase

올바른 순서

(0, 5) → (4, 5) → (3, 4) → (2, 3) → (1, 2) → (1, 6)

1단계 (0, 5): 프림 알고리즘은 시작 노드에서 트리를 확장하면서, **트리에 포함된 노드와 아직 포함되지 않은 노드를 잇는 간선** 중 비용이 가장 작은 것을 선택한다. 노드 0에서 출발하면 후보는 (0, 5)의 비용 10과 (0, 1)의 비용 28이다. 비용 10인 (0, 5)를 선택하여 노드 5를 추가한다.

2단계 (4, 5): 현재 트리에 포함된 노드는 0과 5이다. 바깥으로 연결되는 후보는 (4, 5)의 비용 25와 (0, 1)의 비용 28이므로, (4, 5)를 선택하여 노드 4를 추가한다. 비용 14인 (1, 6)은 **양 끝 노드가 모두 트리 밖에 있어** 이 단계의 후보가 아니다.

3단계 (3, 4): 현재 트리의 노드는 0·5·4이다. 후보인 (3, 4)의 비용 22, (4, 6)의 비용 24, (0, 1)의 비용 28을 비교하여 (3, 4)를 선택한다. 이로써 노드 3이 추가된다.

4단계 (2, 3): 노드 3이 추가되면서 (2, 3)의 비용 12와 (3, 6)의 비용 18도 후보가 된다. 기존 후보인 (4, 6)의 비용 24, (0, 1)의 비용 28까지 비교하면 가장 작은 비용은 12이므로, (2, 3)을 선택하여 노드 2를 추가한다.

5단계 (1, 2): 노드 2가 추가되면 비용 16인 (1, 2)가 새 후보가 된다. 이는 (3, 6)의 비용 18, (4, 6)의 비용 24, (0, 1)의 비용 28보다 작으므로, (1, 2)를 선택하여 노드 1을 추가한다.

6단계 (1, 6): 이제 트리 밖에는 노드 6만 남는다. 노드 6을 잇는 후보는 (1, 6)의 비용 14, (3, 6)의 비용 18, (4, 6)의 비용 24이다. 그중 **비용 14인 (1, 6)**을 선택하면 모든 노드가 연결되므로 이 간선이 마지막에 선택된다.

NoteBase

문 9. 가상기억장치(virtual memory) 구현 방법으로서의 페이징(paging)과 세그멘테이션(segmentation)에 대한 설명으로 옳지 않은 것은?

① 페이징 기법에서 페이지(page)의 크기가 2^k바이트이면 가상 주소(virtual address)의 페이지 오프셋(offset)은 k비트이다.

② 세그멘테이션 기법에서 세그먼트들은 2의 거듭제곱 바이트의 크기를 가져야 하며 최대 크기가 정해져 있다.

③ 페이징 기법에서는 외부 단편화(external fragmentation)가 발생하지 않는다.

④ 세그멘테이션 기법에서는 외부 단편화가 발생할 수 있다.

정답 2번

간단한 해설

세그멘테이션은 프로그램을 논리적 단위에 따라 **가변 길이**의 세그먼트로 나눈다. 세그먼트 크기가 반드시 **2의 거듭제곱일 필요는 없다**.

선지별 해설

① O: 페이지 오프셋은 페이지 내부에서 특정 바이트의 위치를 나타낸다. 페이지 크기가 2^k바이트이면 바이트 위치는 0부터 $2^k - 1$까지이므로, 이를 구별하는 데 **k비트**가 필요하다.

② X: 세그멘테이션은 코드·데이터·스택 등 프로그램의 논리적 단위를 **서로 다른 길이의 세그먼트**로 나누는 방식이다. 각 세그먼트의 크기는 필요한 공간에 따라 정하므로 2의 거듭제곱 바이트로 제한되지 않는다. 주소 형식 등 구현에 따라 세그먼트의 최대 크기가 정해질 수 있지만, 이것이 실제 세그먼트 크기도 반드시 2의 거듭제곱이어야 한다는 뜻은 아니다.

③ O: 외부 단편화는 빈 공간이 여러 곳에 흩어져 있어 필요한 연속 공간을 확보하지 못하는 현상이다. 페이징에서는 물리 메모리도 페이지와 같은 크기의 프레임으로 나누고, 각 페이지를 **서로 떨어진 빈 프레임에도 배치**할 수 있다. 따라서 전체 프로그램을 위한 연속 공간을 확보할 필요가 없어 외부 단편화가 발생하지 않는다.

④ O: 세그멘테이션에서는 크기가 서로 다른 각 세그먼트를 **하나의 연속된 메모리 영역**에 배치한다. 세그먼트의 할당과 해제가 반복되면 빈 공간이 여러 곳에 흩어질 수 있다. 이때 빈 공간의 총합이 충분해도 새 세그먼트를 넣을 만큼 큰 연속 공간이 없으면 배치할 수 없으므로 외부 단편화가 발생할 수 있다.

NoteBase

문 10. 캐시 일관성(cache coherence) 문제를 해결하기 위한 기술과 관련이 없는 것은?

① 스누핑(snooping) 프로토콜

② MESI 프로토콜

③ 디렉토리 기반(directory-based) 프로토콜

④ 우선순위 상속(priority-inheritance) 프로토콜

정답 4번

간단한 해설

우선순위 상속 프로토콜은 자원을 보유한 작업의 실행 우선순위를 일시적으로 높여 **우선순위 역전을 완화**한다. 스누핑·MESI·디렉토리 기반 프로토콜처럼 여러 캐시의 데이터 사본을 일관되게 관리하는 기술이 아니다.

개념 구분

캐시 일관성은 여러 프로세서가 같은 메모리 위치의 데이터를 캐시에 보관할 때, 쓰기로 변경된 값이 다른 프로세서에도 일관되게 관찰되도록 하는 성질이다. 캐시 일관성 프로토콜은 다른 캐시의 오래된 사본을 무효화하거나 갱신하는 등의 방식으로 이를 유지한다.

① 스누핑 프로토콜: 각 캐시 제어기가 **공유 버스 등에 전달되는 일관성 관련 요청을 감시**한다. 다른 프로세서의 요청이 자신이 보관한 데이터와 관련되면 사본을 무효화하거나 필요한 데이터를 제공하여 일관성을 유지한다.

② MESI 프로토콜: 캐시 라인을 수정(Modified), 배타(Exclusive), 공유(Shared), 무효(Invalid)의 **네 상태로 관리**한다. 읽기·쓰기 요청에 따라 상태를 바꾸며, 공유 중인 데이터를 쓰려면 다른 캐시의 사본을 무효화하는 등 캐시 사이의 접근을 조정한다.

③ 디렉토리 기반 프로토콜: 메모리 블록별로 **어느 캐시가 사본을 보유하는지** 등의 정보를 디렉토리에 기록한다. 이 정보를 이용하여 관련 캐시에 무효화 요청 등 필요한 메시지를 보내므로 캐시 일관성 유지 기술에 해당한다.

④ 우선순위 상속 프로토콜: 높은 우선순위의 작업이 낮은 우선순위의 작업이 보유한 자원을 기다리면, **자원 보유 작업이 기다리는 작업의 높은 우선순위를 일시적으로 상속**한다. 그 결과 중간 우선순위의 작업이 자원 보유 작업의 실행을 지연시키는 것을 막아 우선순위 역전을 완화한다. 관리 대상은 작업의 실행 우선순위이다.

따라서 캐시 일관성 문제 해결과 관련이 없는 것은 ④ 우선순위 상속 프로토콜이다.

NoteBase

문 11. 통신 연결 장치와 그 장치가 동작하는 OSI(Open Systems Interconnection) 계층이 바르게 짝지어진 것은?

ㄱ. 네트워크 계층(network layer)
ㄴ. 데이터 링크 계층(data link layer)
ㄷ. 물리 계층(physical layer)

① 라우터(router): ㄱ, 브리지(bridge): ㄴ, 리피터(repeater): ㄷ
② 라우터(router): ㄴ, 브리지(bridge): ㄱ, 리피터(repeater): ㄷ
③ 라우터(router): ㄴ, 브리지(bridge): ㄷ, 리피터(repeater): ㄱ
④ 라우터(router): ㄷ, 브리지(bridge): ㄴ, 리피터(repeater): ㄱ

정답 1번

간단한 해설

라우터의 **IP 주소를 이용한 경로 선택**은 네트워크 계층, 브리지의 **MAC 주소를 이용한 프레임 전달**은 데이터 링크 계층의 기능이다. 리피터는 주소를 판단하지 않고 신호를 재생하므로 물리 계층에 해당한다.

대응 관계 해설

라우터(router) → ㄱ. 네트워크 계층(3계층): 서로 다른 네트워크 사이에서 패킷을 전달한다. 패킷의 **목적지 IP 주소와 라우팅 테이블**을 이용하여 다음 전달 경로를 결정하므로 네트워크 계층 장치이다.

브리지(bridge) → ㄴ. 데이터 링크 계층(2계층): LAN 구간을 연결하며 프레임을 전달하거나 걸러 낸다. 프레임의 **목적지 MAC 주소**를 확인하여 어느 구간으로 전달할지 판단하므로 데이터 링크 계층 장치이다.

리피터(repeater) → ㄷ. 물리 계층(1계층): 전송 중 약해지거나 변형된 신호를 재생하여 전송 거리를 늘린다. **주소나 프레임의 내용을 해석하지 않고 신호 자체를 처리**하므로 물리 계층 장치이다.

따라서 라우터–ㄱ, 브리지–ㄴ, 리피터–ㄷ으로 대응하며, 이는 ①의 조합과 일치한다.

NoteBase

문 12. 교착상태(deadlock)가 발생하기 위해서 만족해야 하는 조건들에 대한 설명으로 옳지 않은 것은?

① 상호 배제(mutual exclusion) 조건: 한 프로세스에 의해 점유된 자원은 다른 프로세스가 사용할 수 없다.

② 점유와 대기(hold and wait) 조건: 이미 하나 이상의 자원을 점유한 프로세스가 다른 프로세스에 의해 점유된 자원을 요청하며 대기하고 있다.

③ 비선점(no preemption) 조건: 프로세스가 점유한 자원을 그 프로세스로부터 강제로 빼앗을 수 있다.

④ 순환 대기(circular wait) 조건: 프로세스 간에 닫힌 체인(closed chain)이 존재하여, 체인 내의 각 프로세스는 체인 내의 다른 프로세스에 의해 소유되어 있는 자원을 요청하며 대기하고 있다.

정답 3번

간단한 해설

비선점 조건은 프로세스가 점유한 자원을 **강제로 빼앗을 수 없다**는 뜻이다. ③의 '강제로 빼앗을 수 있다'는 비선점과 반대되는 설명이다.

선지별 해설

교착상태는 프로세스들이 서로 필요한 자원을 기다려 더 이상 진행하지 못하는 상태이다. 교착상태가 발생하려면 다음 네 가지 필요조건이 동시에 성립해야 한다.

① O: 상호 배제는 하나의 자원을 **동시에 여러 프로세스가 사용할 수 없는** 조건이다. 한 프로세스가 자원을 점유하여 사용하는 동안 다른 프로세스는 그 자원을 사용하려면 기다려야 한다.

② O: 점유와 대기는 프로세스가 **기존 자원을 점유한 채** 다른 프로세스가 점유한 추가 자원을 요청하며 기다리는 조건이다. 필요한 자원을 기다리는 동안에도 이미 확보한 자원을 반납하지 않는다는 점이 핵심이다.

③ X: 비선점은 프로세스가 점유한 자원을 **강제로 빼앗을 수 없는** 조건이다. 자원은 점유한 프로세스가 스스로 반납해야 사용할 수 있다. 따라서 '강제로 빼앗을 수 있다'를 '강제로 빼앗을 수 없다'로 고쳐야 한다.

④ O: 순환 대기는 프로세스들의 자원 대기 관계가 **고리를 이루는** 조건이다. 예를 들어 A가 자원 R1을 점유하고 B의 자원 R2를 기다리는 동시에, B가 R2를 점유하고 A의 R1을 기다리면 대기 관계가 A → B → A로 닫힌다. 이는 각 프로세스가 고리 안의 다른 프로세스가 점유한 자원을 기다리는 상태이다.

NoteBase

문 13. 자료 구조에 대한 설명으로 옳지 않은 것은?

① 큐(queue)는 선입 선출의 특성을 가지며 삽입과 삭제가 서로 다른 끝 쪽에서 일어난다.

② 연결 그래프(connected graph)에서는 그래프 내의 모든 노드 간에 갈 수 있는 경로가 존재한다.

③ AVL 트리는 삽입 또는 삭제가 일어나 트리의 균형이 깨지는 경우 트리 모습을 변형시킴으로써 균형을 복원시킨다.

④ 기수 정렬(radix sort)은 키(key) 값이 가장 큰 것과 가장 오른쪽 것의 위치 교환을 반복적으로 수행한다.

정답 4번

간단한 해설

④의 '가장 큰 키와 가장 오른쪽 원소를 교환한다'는 설명은 선택 정렬의 한 방식이다. 기수 정렬은 키 전체의 대소를 비교하는 대신 **각 자릿값에 따라 분류하고 모으는 과정**으로 정렬한다.

선지별 해설

① O: 큐는 먼저 넣은 자료가 먼저 나오는 **선입선출** 구조이다. 뒤쪽(rear)에서 삽입하고 앞쪽(front)에서 삭제한다. 예를 들어 A를 넣고 B를 넣으면 A가 먼저 나온다.

② O: 연결 그래프는 **임의의 두 노드 사이에 경로가 존재**하는 그래프이다. 모든 노드 쌍이 간선 하나로 직접 연결되어야 한다는 뜻은 아니며, 다른 노드를 거쳐 도달할 수 있어도 된다.

③ O: AVL 트리는 각 노드에서 왼쪽과 오른쪽 서브트리의 높이 차이를 1 이하로 유지하는 균형 이진 탐색 트리이다. 삽입이나 삭제로 이 조건이 깨지면 **회전 연산으로 균형을 복원**한다. 이때 이진 탐색 트리의 키 대소 관계는 유지된다.

④ X: 기수 정렬은 키를 구성하는 **자릿값별로 분류하고 모아** 정렬하는 방법이다. 예를 들어 낮은 자리부터 처리하는 방식은 같은 자릿값을 가진 원소의 기존 순서를 유지하면서 일의 자리, 십의 자리 순으로 정렬한다. 미정렬 구간의 최댓값을 찾아 가장 오른쪽 원소와 교환하고 그 원소를 정렬 대상에서 제외하는 과정은 **선택 정렬의 한 방식**이다.

NoteBase

문 14. 단일 종류의 논리 게이트(gate)만을 사용하더라도 모든 조합논리 회로를 구현할 수 있는 게이트로 옳은 것은?

① AND 게이트

② OR 게이트

③ NOR 게이트

④ 인버터(inverter)

정답 3번

간단한 해설

NOR 게이트는 여러 개를 연결하면 NOT·OR·AND 연산을 모두 구현할 수 있는 **범용 게이트**이다. 따라서 다른 종류의 게이트를 섞지 않아도 모든 조합논리 회로를 구성할 수 있다.

공통 해설

한 종류의 게이트만으로 모든 조합논리 회로를 구현할 수 있으면 그 게이트를 '범용 게이트'라고 한다. 모든 조합논리 함수는 AND·OR·NOT의 조합으로 표현할 수 있으므로, **한 종류만으로 이 세 연산을 모두 구현할 수 있는지**가 판단 기준이다. '단일 종류'는 게이트를 한 개만 쓴다는 뜻이 아니라, 같은 종류의 게이트를 여러 개 연결해도 된다는 뜻이다.

NOR는 OR의 결과를 반전하는 게이트로, 입력이 모두 0일 때만 1을 출력한다. NOR의 두 입력에 같은 신호 A를 넣으면 A가 0일 때 1, A가 1일 때 0이 출력되어 NOT을 구현한다. 두 신호의 NOR 출력에 이렇게 만든 NOT을 연결하면 한 번 더 반전되므로 OR가 된다. 또한 두 입력을 각각 NOT으로 반전시킨 뒤 NOR에 넣으면, 원래 두 입력이 모두 1일 때만 출력이 1이 되어 AND가 된다. 이처럼 **NOR만으로 NOT·OR·AND를 모두 구성**할 수 있으므로 모든 조합논리 회로를 구현할 수 있다.

AND만 연결하거나 OR만 연결하면 입력을 반전하는 NOT을 구현할 수 없다. 인버터만 연결하면 개별 신호를 반전할 수는 있지만, 서로 다른 두 입력을 결합하는 AND·OR를 구현할 수 없다. 따라서 이들은 각각 단독으로 사용할 때 모든 조합논리 회로를 구현하는 조건을 충족하지 못한다.

NoteBase

문 15. 데이터베이스 데이터 모델에 대한 설명으로 옳지 않은 것은?

① 계층 데이터 모델은 트리 형태의 데이터 구조를 가진다.

② 관계 데이터 모델은 테이블로 데이터베이스를 나타낸다.

③ 네트워크 데이터 모델은 그래프 형태로 데이터베이스 구조를 표현한다.

④ 계층 데이터 모델, 관계 데이터 모델, 네트워크 데이터 모델은 개념적 데이터 모델이다.

정답 4번

간단한 해설

계층·관계·네트워크 데이터 모델은 데이터를 트리·테이블·그래프 등의 구조로 표현하는 **논리적 데이터 모델**이다. 개념적 데이터 모델은 현실의 대상과 관계를 추상화하는 모델로, 대표적으로 E-R 모델이 있다.

선지별 해설

① O: 계층 데이터 모델은 부모와 자식 레코드를 **트리 구조**로 연결한다. 루트를 제외한 각 레코드는 하나의 부모를 가지며, 부모와 자식 사이의 일대다 관계를 표현한다.

② O: 관계 데이터 모델은 데이터를 행과 열로 이루어진 **테이블(릴레이션)**로 표현한다. 행은 하나의 레코드에 해당하는 튜플이고, 열은 데이터의 특성을 나타내는 속성이다.

③ O: 네트워크 데이터 모델은 레코드와 레코드 사이의 연결을 **그래프 구조**로 표현한다. 계층 모델과 달리 한 레코드가 여러 부모 레코드와 연결될 수 있어 다대다 관계를 표현할 수 있다.

④ X: 계층·관계·네트워크 데이터 모델은 모두 **논리적 데이터 모델**이다. 데이터베이스에서 데이터를 어떤 논리적 구조로 표현할지 정하기 때문이다. '개념적 데이터 모델'은 '논리적 데이터 모델'로 고쳐야 한다. 개념적 데이터 모델은 특정 데이터베이스 구현 방식에 얽매이지 않고 현실의 개체·속성·관계를 추상화하며, E-R 모델이 대표적이다.

NoteBase

문 16. 파이프라이닝(pipelining) 기법이 적용된 중앙처리장치(CPU)에서의 파이프라인 해저드(pipeline hazard) 종류와 대응 방법을 바르게 짝지은 것만을 모두 고른 것은?

ㄱ. 데이터 해저드(data hazard) - 데이터 전방전달(data forwarding)
ㄴ. 구조적 해저드(structural hazard) - 부족한 자원의 추가
ㄷ. 제어 해저드(control hazard) - 분기 예측(branch prediction)

① ㄱ, ㄴ
② ㄱ, ㄷ
③ ㄴ, ㄷ
④ ㄱ, ㄴ, ㄷ

정답 4번

간단한 해설

분기 명령은 다음에 실행할 명령의 주소를 불확실하게 하므로 **제어 해저드**를 일으킨다. **분기 예측**은 분기 결과가 확정되기 전에 실행할 경로를 예상하여 명령어를 가져옴으로써 지연을 줄이는 방법이다.

NoteBase

보기별 해설

파이프라이닝은 명령어 처리를 여러 단계로 나누어 서로 다른 명령어의 단계를 겹쳐 실행하는 기법이다. 해저드는 이러한 중첩 실행 중 다음 단계의 정상적인 진행을 방해하는 상황이다.

ㄱ. O: 데이터 해저드는 앞 명령의 처리 결과를 뒤 명령이 필요로 하지만 그 결과가 아직 준비되지 않았거나 전달되지 않아 발생한다. 데이터 전방전달은 앞 명령의 결과를 레지스터에 기록할 때까지 기다리지 않고 **그 결과를 필요한 실행 단계로 직접 전달**하는 방법이다. 예를 들어 앞 명령의 덧셈 결과를 바로 다음 명령이 사용할 때 대기 시간을 줄일 수 있다. 다만 결과가 필요한 시점까지 생성되지 않았다면 전방전달을 사용해도 대기가 필요할 수 있다.

ㄴ. O: 구조적 해저드는 동시에 실행 중인 명령들이 **같은 하드웨어 자원을 동시에 요구**하지만 그 자원이 이를 처리하지 못할 때 발생한다. 예를 들어 명령어 인출과 데이터 접근이 하나의 메모리 접근 자원을 두고 충돌할 수 있다. 이때 명령어용과 데이터용 접근 자원을 분리하거나 부족한 자원을 추가하면 자원 충돌을 줄일 수 있다.

ㄷ. O: 제어 해저드는 분기 결과나 목적지가 확정되지 않아 다음에 가져올 명령어를 결정하기 어려울 때 발생한다. 분기 예측은 **분기 결과가 확정되기 전에 실행 경로를 예상**하여 명령어 인출과 실행을 이어 가는 방법이다. 예측이 맞으면 대기를 줄일 수 있고, 틀리면 잘못된 경로에서 처리한 명령을 폐기하고 올바른 경로로 진행해야 한다.

ㄱ·ㄴ·ㄷ의 짝이 모두 적절하므로, 세 보기를 모두 포함한 ④에 해당한다.

NoteBase

문 17. 유비쿼터스 컴퓨팅에 대한 설명으로 옳지 않은 것은?

① 감지 컴퓨팅은 컴퓨터가 센서 등을 이용하여 사용자의 행위 또는 주변 환경을 인식하여 필요 정보를 제공하는 기술이다.

② 노매딕(nomadic) 컴퓨팅은 현실 세계와 가상 화면을 결합하여 보여주는 기술이다.

③ 퍼베이시브(pervasive) 컴퓨팅은 컴퓨터가 도처에 편재되도록 하는 기술이다.

④ 웨어러블(wearable) 컴퓨팅은 컴퓨터 착용을 통해 컴퓨터를 인간 몸의 일부로 여길 수 있도록 하는 기술이다.

정답 2번

간단한 해설

②의 '현실 세계와 가상 화면을 결합'하는 기술은 **증강현실(AR)**이다. 노매딕 컴퓨팅은 사용자가 **장소를 옮겨도 컴퓨팅 서비스를 이용**할 수 있도록 지원하는 기술이다.

선지별 해설

① O: 감지 컴퓨팅은 센서로 사용자의 행동이나 주변 환경의 정보를 수집하고, 이를 바탕으로 **상황에 맞는 정보나 서비스**를 제공한다. 예를 들어 주변 밝기를 감지해 화면 밝기를 조절하는 방식이다.

② X: 노매딕 컴퓨팅은 사용자가 여러 장소로 이동하면서도 컴퓨터와 네트워크를 통해 필요한 서비스를 이용할 수 있도록 지원한다. 핵심은 **사용자의 이동에 따른 컴퓨팅 이용 지원**이다. 현실의 영상에 가상의 정보나 이미지를 겹쳐 보여주는 것은 **증강현실(AR)**이므로, 선지는 서로 다른 기술을 연결했다.

③ O: 퍼베이시브 컴퓨팅은 컴퓨터가 특정 장소에만 있는 것이 아니라 **생활 공간 곳곳의 사물과 환경에 스며들도록** 하는 기술이다. 여기서 '편재'는 도처에 널리 존재한다는 뜻이다.

④ O: 웨어러블 컴퓨팅은 컴퓨팅 장치를 옷이나 안경처럼 **몸에 착용하여 이용**하는 기술이다. 장치를 신체 활동과 밀접하게 결합해 사용한다는 점에서 컴퓨터를 몸의 일부처럼 활용하도록 한다.

NoteBase

문 18. 다음 C 프로그램의 출력 값은?

```
#include <stdio.h>
int main() {
    int a[] = {1, 2, 4, 8};
    int *p = a;

    p[1] = 3;
    a[1] = 4;
    p[2] = 5;

    printf("%d, %d\n", a[1]+p[1], a[2]+p[2]);

    return 0;
}
```

① 5, 9
② 6, 9
③ 7, 9
④ 8, 10

정답 4번

간단한 해설

포인터 p는 배열 a의 첫 원소를 가리키므로 **p[1]과 a[1]은 같은 원소**이다. p[1]에 3을 저장한 뒤 a[1]에 4를 저장하면 둘 다 4가 되고, p[2]에 5를 저장하면 a[2]도 5가 되어 출력은 8, 10이 된다.

NoteBase

계산 과정

초기 상태: 배열 a의 원소는 인덱스 0부터 차례로 1, 2, 4, 8이다. int *p = a;에서 p는 a[0]의 주소를 저장한다. 따라서 **p[i]와 a[i]는 같은 배열 원소**를 나타내며, 별도의 배열이 만들어지는 것은 아니다.

p[1] = 3; 실행: p[1]은 a[1]과 같은 원소이므로 a[1]의 값이 2에서 3으로 바뀐다. 이때 배열은 1, 3, 4, 8이다.

a[1] = 4; 실행: 같은 원소에 다시 값을 대입하므로 **앞서 저장한 3은 4로 덮어써진다**. 이제 a[1]과 p[1]로 읽는 값은 모두 4이고, 배열은 1, 4, 4, 8이다.

p[2] = 5; 실행: p[2]는 a[2]와 같은 원소이므로 a[2]가 4에서 5로 바뀐다. 최종 배열은 1, 4, 5, 8이며 a[2]와 p[2]로 읽는 값은 모두 5이다.

출력 계산: 첫 번째 식 a[1]+p[1]은 4 + 4 = 8이고, 두 번째 식 a[2]+p[2]는 5 + 5 = 10이다. printf는 두 정수를 쉼표와 공백으로 구분하여 출력하므로 출력 값은 8, 10이며, ④의 값과 일치한다.

NoteBase

문 19. TCP/IP 프로토콜에서 TCP 및 UDP에 대한 설명으로 옳지 않은 것은?

① TCP와 UDP는 전송 계층(transport layer)의 프로토콜이다.

② UDP는 중복 전달 및 전송 오류를 허용한다.

③ TELNET, SNMP, TFTP는 TCP 서비스를 이용하는 응용 계층(application layer) 프로토콜이다.

④ TCP는 신뢰성 있는 통신을 제공하기 위한 연결형 프로토콜이다.

정답 3번

간단한 해설

TELNET은 TCP를 이용하지만, **SNMP와 TFTP는 기본적으로 UDP**를 이용한다. 세 프로토콜이 모두 TCP 서비스를 이용한다는 설명은 전송 계층 프로토콜의 대응을 잘못 제시한 것이다.

선지별 해설

① O: TCP와 UDP는 모두 **전송 계층** 프로토콜이다. IP가 호스트 사이의 패킷 전달을 담당하는 데 비해, TCP와 UDP는 포트 번호를 이용하여 데이터를 해당 호스트의 응용 프로세스로 전달한다.

② O: UDP는 비연결형 프로토콜로, **중복 제거와 재전송에 의한 오류 복구를 자체적으로 보장하지 않는다**. 따라서 데이터가 중복되거나 손실되어도 UDP 자체가 이를 복구하는 것은 아니다. 선지의 '전송 오류를 허용한다'는 신뢰성 있는 전달을 보장하지 않는다는 뜻이며, 체크섬을 통한 오류 검출 기능이 없다는 뜻은 아니다. 필요한 복구 기능은 응용 계층에서 별도로 구현할 수 있다.

③ X: TELNET, SNMP, TFTP는 모두 응용 계층 프로토콜이지만, 이용하는 전송 계층 프로토콜은 같지 않다. 원격 터미널 접속에 쓰이는 TELNET은 TCP를 이용하고, 네트워크 관리에 쓰이는 **SNMP와 간단한 파일 전송에 쓰이는 TFTP는 기본적으로 UDP**를 이용한다. 따라서 세 프로토콜이 모두 TCP를 이용한다는 부분을 고쳐야 한다.

④ O: TCP는 데이터를 주고받기 전에 연결을 설정하는 **연결형 프로토콜**이다. 순서 번호와 확인 응답으로 전달 상태를 관리하고, 손실된 데이터 등을 재전송하여 응용 프로그램에 신뢰성 있고 순서가 유지되는 바이트 스트림을 제공한다.

NoteBase

문 20. 인터넷에서 사용되는 경로배정(routing) 프로토콜 중에서 자율 시스템(autonomous system) 내부에서의 경로배정을 위해 사용되는 것만을 모두 고른 것은?

ㄱ. OSPF
ㄴ. BGP
ㄷ. RIP

① ㄱ, ㄴ
② ㄱ, ㄷ
③ ㄴ, ㄷ
④ ㄱ, ㄴ, ㄷ

정답 2번

간단한 해설

BGP는 자율 시스템 간에 경로 정보를 교환하기 위한 **외부 경로배정 프로토콜(EGP)**로 분류된다. 문제에서 묻는 것은 자율 시스템 내부의 경로배정을 담당하는 IGP이다.

보기별 해설

자율 시스템(AS)은 하나의 관리 주체 아래 공통된 경로배정 정책으로 운영되는 네트워크 집합이다. 경로배정 프로토콜은 AS 내부에서 사용하는 IGP와 AS 간 경로 정보 교환에 사용하는 EGP로 구분한다.

ㄱ. OSPF: **내부 경로배정 프로토콜(IGP)**이다. AS 내부의 링크 상태 정보를 공유하여 네트워크 구성을 파악하고, 링크 비용을 기준으로 최단 경로를 구한다.

ㄴ. BGP: **외부 경로배정 프로토콜(EGP)**로 분류된다. 서로 다른 AS 사이에서 경로 정보를 교환하며, 경유하는 AS 정보와 경로 정책 등을 이용해 경로를 선택한다. 따라서 문제에서 묻는 IGP에는 포함하지 않는다.

ㄷ. RIP: **내부 경로배정 프로토콜(IGP)**이다. 거리 벡터 방식으로 이웃 라우터와 경로 정보를 교환하며, 목적지까지 거치는 라우터 수인 홉 수를 경로 선택의 기준으로 사용한다.

AS 내부 경로배정에 사용하는 것은 ㄱ과 ㄷ이므로 ②의 조합에 해당한다.

NoteBase