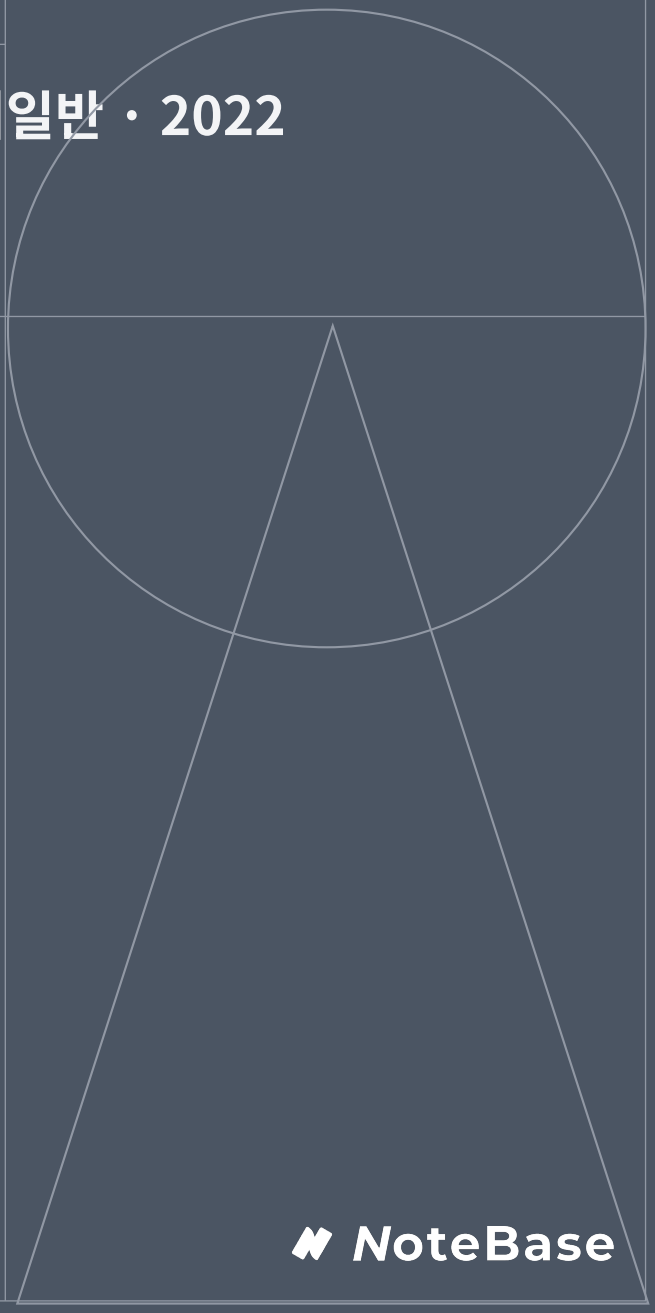
기출OX 해설집

공무원/계리직/9급 · 컴퓨터일반 · 2022



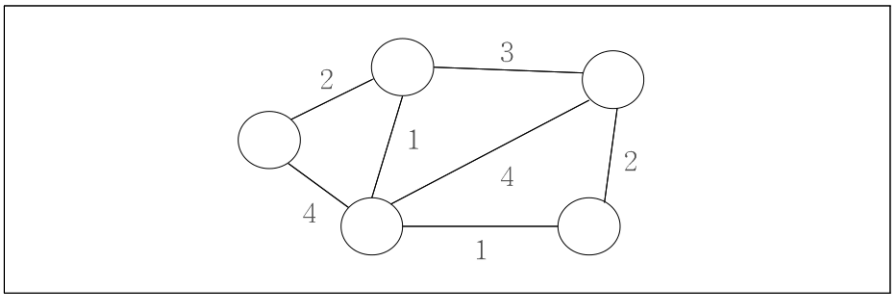
NoteBase

기출OX 해설집

공무원/계리직/9급 · 컴퓨터일반 · 2022

공무원/계리직/9급 · 컴퓨터일반 · 2022

문 1. 다음 가중치 그래프에서 최소 비용 신장 트리(minimum cost spanning tree)의 가중치 합은?



① 4 ② 6 ③ 13 ④ 17

정답 2번

출제 포인트

최소 비용 신장 트리는 그래프의 모든 정점을 사이클 없이 연결하면서 간선 가중치의 합이 최소가 되는 트리이다. 이 도식에서는 가중치가 작은 간선부터 선택하되 사이클이 생기는 간선은 제외하여 최소 합을 구한다.

NoteBase

선지별 해설

① X
- 4는 정점 5개로 이루어진 신장 트리에 필요한 간선의 개수이지 가중치의 합이 아니다. 중앙 위-중앙 아래의 1, 중앙 아래-오른쪽 아래의 1, 왼쪽-중앙 위의 2, 오른쪽 위-오른쪽 아래의 2를 선택하면 가중치 합은 6이다. 따라서 올바르게는 6이다.

② O
- 가중치 1인 두 간선과 가중치 2인 두 간선을 선택하면 5개 정점이 모두 연결되고 사이클이 생기지 않는다. 합은 1+1+2+2=6이므로 최소 비용 신장 트리의 가중치 합은 6이다.

③ X
- 13은 최소 가중치 간선부터 사이클 없이 고른 결과가 아니다. 정점 5개를 연결하는 데 필요한 네 간선을 1, 1, 2, 2로 선택할 수 있으므로 올바르게는 6이다.

④ X
- 17은 그림에 있는 모든 간선의 가중치 2+4+1+3+4+1+2를 더한 값이다. 신장 트리는 모든 간선을 사용하는 것이 아니라 정점 5개를 연결하는 네 간선만 사용하므로 올바르게는 6이다.

오답 포인트

신장 트리의 간선 수인 4를 가중치 합으로 착각하거나, 그림의 모든 간선 가중치를 더해 17로 판단하기 쉽다. 이 문제에서는 간선 수와 가중치 합을 구분하고, 작은 가중치부터 선택하면서 사이클을 제외해야 한다.

다시 나오면 이렇게 판단

정점이 5개이므로 신장 트리는 반드시 4개의 간선을 가진다. 가중치가 작은 순서대로 간선을 검토하고, 사이클을 만들지 않으면서 모든 정점이 연결될 때까지 선택한 뒤 그 가중치만 합산한다.

암기 포인트

신장 트리의 간선 수는 정점 수-1이고, 최소 비용 신장 트리는 작은 가중치부터 사이클 없이 선택한다.

NoteBase

문 2. 다음은 정렬 알고리즘을 이용해 초기 단계의 데이터를 완료 단계의 데이터로 정렬하는 과정을 보여 준다. 이 과정에 사용된 정렬 알고리즘으로 적절한 것은?

단 계	데 이 터					
초기	534	821	436	773	348	512
1	821	512	773	534	436	348
2	512	821	534	436	348	773
완료	348	436	512	534	773	821

① 기수(radix) 정렬 ② 버블(bubble) 정렬
③ 삽입(insertion) 정렬 ④ 선택(selection) 정렬

정답 1번

출제 포인트

이 문제는 각 단계에서 어느 자릿수를 기준으로 데이터의 순서가 바뀌는지 보고 정렬 알고리즘을 판별하는 문제이다.

올바른 순서

초기: 534, 821, 436, 773, 348, 512
1단계: 일의 자리를 오름차순으로 안정 정렬하여 821, 512, 773, 534, 436, 348이 된다.
2단계: 십의 자리를 오름차순으로 안정 정렬하여 512, 821, 534, 436, 348, 773이 된다.
완료: 백의 자리를 오름차순으로 안정 정렬하여 348, 436, 512, 534, 773, 821이 된다.
왜 이 순서: 가장 낮은 일의 자리부터 십의 자리, 백의 자리 순으로 처리하고, 각 단계에서 같은 자릿값을 가진 데이터의 기존 순서를 유지해야 앞서 정렬한 낮은 자리의 순서가 보존되기 때문이다.

NoteBase

선지별 해설

① O — 기수(radix) 정렬이다. 일의 자리, 십의 자리, 백의 자리 순으로 안정 정렬한 결과가 표의 1단계, 2단계, 완료 단계와 정확히 일치한다.
② X — 틀린 부분은 '버블(bubble) 정렬'이다. 버블 정렬은 인접한 두 값을 비교하여 역순이면 교환하며, 한 차례 진행하면 큰 값이 뒤쪽으로 이동한다. 표처럼 자릿수별로 821, 512, 773, 534, 436, 348을 만드는 방식이 아니다. 올바르게는 기수(radix) 정렬이다.
③ X — 틀린 부분은 '삽입(insertion) 정렬'이다. 삽입 정렬은 앞부분을 정렬된 구간으로 두고 다음 원소를 알맞은 위치에 삽입한다. 표처럼 전체 데이터를 일의 자리부터 묶어 재배치하지 않는다. 올바르게는 각 자릿수를 낮은 자리부터 차례로 안정 정렬하는 기수(radix) 정렬이다.
④ X — 틀린 부분은 '선택(selection) 정렬'이다. 선택 정렬은 정렬되지 않은 구간에서 최솟값을 선택하여 맨 앞에 놓으므로, 오름차순의 첫 단계라면 최솟값 348이 앞쪽에 놓여야 한다. 표의 1단계 첫 값은 821이므로 선택 정렬이 아니다. 올바르게는 기수(radix) 정렬이다.

오답 포인트

중간 결과가 아직 전체 크기순으로 정렬되지 않았다는 이유만으로 잘못된 과정이라고 판단하기 쉽다. 이 표에서는 1단계의 순서 821, 512, 773, 534, 436, 348이 각 수의 일의 자리 1, 2, 3, 4, 6, 8의 순서라는 점이 결정적 단서이다.

다시 나오면 이렇게 판단

단계별 배열에서 각 수의 일의 자리, 십의 자리, 백의 자리를 차례로 대조한다. 낮은 자리부터 자릿값 순으로 재배치되고 같은 자릿값끼리 기존 순서가 유지되면 기수 정렬로 판단한다.

암기 포인트

기수 정렬은 낮은 자리부터 높은 자리까지, 같은 자릿값의 순서는 그대로 유지한다.

NoteBase

문 3. 노드의 수가 60개인 이진 트리의 최대 높이에서 최소 높이를 뺀 값은?

① 53 ② 54 ③ 55 ④ 56

정답 2번

출제 포인트

노드가 60개인 이진 트리의 최대 높이와 최소 높이를 각각 구한 뒤 그 차를 계산하는 문제이다.

조건 정리

주어진 값: 노드 수는 60개이다.
구해야 할 값: 최대 높이에서 최소 높이를 뺀 값이다.
주의할 조건: 높이를 간선 수로 세든 레벨 수로 세든 두 높이에 같은 기준을 적용해야 한다.

사용할 공식

높이를 루트에서 가장 먼 리프까지의 간선 수 h로 정의하면, 높이 h인 이진 트리의 최대 노드 수는 $2^{h+1}-1$이다. 따라서 최소 높이는 $2^{h+1}-1$이 60 이상이 되는 가장 작은 h이고, 한쪽 자식으로만 이어진 이진 트리의 최대 높이는 노드 수가 n일 때 $n-1$이다. 완전하게 채울수록 높이가 작아지고 한 줄로 치우칠수록 높이가 커지므로 이 공식을 사용한다.

계산 과정

1. 최소 높이를 구하면 $2^5-1=31<60$이고 $60\le 2^6-1=63$이므로 $h_{\min}=5$이다.
2. 최대 높이는 60개 노드가 한쪽으로만 이어질 때이므로 $h_{\max}=60-1=59$이다.
3. 따라서 높이의 차는 $h_{\max}-h_{\min}=59-5=54$이다.

높이를 레벨 수로 정의하면 최소 6, 최대 60이 되어 차는 역시 54이다.

선지 판단

① X — 53은 최대 높이를 간선 수인 59로 세면서 최소 높이는 레벨 수인 6으로 세어 기준을 섞은 값이다. 같은 기준으로 계산하면 53이 아니라 54이다.
② O — 최대 높이 59에서 최소 높이 5를 빼면 54이므로 계산 결과와 일치한다.
③ X — 55는 최대 높이를 레벨 수인 60으로 세면서 최소 높이는 간선 수인 5로 세어 기준을 섞은 값이다. 같은 기준으로 계산하면 55가 아니라 54이다.
④ X — 60개 노드는 간선 기준 높이 4인 이진 트리에 들어갈 수 없다. 높이 4에서 가능한 최대 노드 수는 $2^5-1=31$개이므로 최소 높이는 4가 아니라 5이며, 올바른 차는 56이 아니라 54이다.

NoteBase

오답 포인트

높이를 간선 수로 세는 기준과 레벨 수로 세는 기준을 섞으면 53이나 55가 나온다. 최대 높이와 최소 높이에 동일한 기준을 적용하면 기준에 관계없이 차는 54이다.

다시 나오면 이렇게 판단

노드 수가 n인 이진 트리에서는 한 줄로 치우친 경우로 최대 높이 $n-1$을 구하고, $2^{h+1}-1 \geq n$을 만족하는 가장 작은 h로 최소 높이를 구한 뒤 두 값을 뺀다.

암기 포인트

최대 높이는 한 줄 트리, 최소 높이는 꽉 찬 트리로 판단한다. 높이 기준은 끝까지 통일한다.

NoteBase

문 4. <보기>에서 TCP에 대한 설명으로 옳은 것을 모두 고른 것은?

<보 기>

ㄱ. RTT(Round Trip Time) 측정이 필요하다.
ㄴ. 하나의 TCP 연결로 양방향 데이터 전달이 가능하다.
ㄷ. 라우터 혼잡을 피하기 위해 흐름 제어(flow control)를 수행한다.
ㄹ. TCP 헤더(옵션 제외)에 데이터의 길이 정보를 나타내는 길이 필드(length field)가 존재한다.
ㅁ. 순서(sequence) 번호와 확인(acknowledgement) 번호를 사용한다.

① ㄱ, ㄷ ② ㄱ, ㄴ, ㄹ
③ ㄱ, ㄴ, ㅁ ④ ㄴ, ㄷ, ㅁ

정답 3번

출제 포인트

TCP의 연결 특성, 재전송 시간 관리, 흐름 제어의 목적, 헤더 필드의 역할을 정확히 구분하는 문제이다.

핵심 개념

TCP는 연결 지향형·신뢰성 전송 프로토콜로서 하나의 연결에서 양방향 통신을 지원한다. 재전송 타이머를 정하기 위해 RTT를 추정하고, 순서 번호와 확인 번호로 데이터의 순서와 수신 여부를 관리한다. 흐름 제어는 송신 속도를 수신 측의 처리 능력에 맞추는 기능이고, 혼잡 제어는 네트워크 내부의 과부하를 줄이는 기능이다. TCP 기본 헤더에는 데이터 길이 필드가 없으며, 데이터 오프셋 필드는 TCP 헤더의 길이를 나타낸다.

NoteBase

선지별 해설

① X — ㄱ은 옳지만 ㄷ이 틀리므로 옳은 보기의 조합이 아니다. ㄷ의 '라우터 혼잡을 피하기 위해 흐름 제어를 수행한다'가 틀린 표현이다. 올바르게는 흐름 제어는 수신 측 버퍼의 넘침을 막고, 라우터 등 네트워크의 혼잡은 혼잡 제어가 다룬다.
② X — ㄱ과 ㄴ은 옳지만 ㄹ이 틀리므로 옳은 보기의 조합이 아니다. ㄹ의 '데이터의 길이 정보를 나타내는 길이 필드가 존재한다'가 틀린 표현이다. 올바르게는 TCP 기본 헤더에는 데이터 길이 필드가 없고, 데이터 오프셋 필드가 TCP 헤더의 길이를 나타낸다.
③ O — ㄱ, ㄴ, ㅁ이 모두 옳다. TCP는 재전송 시간 설정을 위해 RTT를 측정·추정하고, 하나의 연결에서 양방향 데이터 전송을 지원하며, 순서 번호와 확인 번호를 사용하여 신뢰성 있는 전송을 수행한다.
④ X — ㄴ과 ㅁ은 옳지만 ㄷ이 틀리므로 옳은 보기의 조합이 아니다. ㄷ의 '라우터 혼잡을 피하기 위해 흐름 제어를 수행한다'가 틀린 표현이다. 올바르게는 수신 측의 처리 능력에 맞추는 것은 흐름 제어이고, 네트워크 혼잡을 완화하는 것은 혼잡 제어이다.

오답 포인트

ㄷ에서는 흐름 제어와 혼잡 제어의 대상을 바꾸어 놓았고, ㄹ에서는 TCP의 데이터 오프셋 필드를 데이터 길이 필드로 오인하게 만들었다. 특히 데이터 오프셋은 데이터의 길이가 아니라 TCP 헤더의 길이를 뜻한다.

다시 나오면 이렇게 판단

수신 측 버퍼와 처리 속도가 문제이면 흐름 제어, 라우터와 네트워크 과부하가 문제이면 혼잡 제어로 판단한다. TCP 헤더에서는 데이터 오프셋이 헤더 길이를 나타낸다는 점을 확인한 뒤, RTT·양방향 통신·순서 번호·확인 번호를 신뢰성 전송 기능과 연결한다.

암기 포인트

흐름 제어는 수신자 보호, 혼잡 제어는 네트워크 보호, 데이터 오프셋은 TCP 헤더 길이이다.

NoteBase

문 5. 이메일 서비스에서 사용되는 프로토콜로 적절하지 않은 것은?

① DNS ② HTTP ③ RTP ④ TCP

정답 3번

출제 포인트

이 문제는 이메일 서비스를 구성하거나 지원하는 프로토콜과 실시간 멀티미디어 전송 프로토콜을 구별하는 문제이다.

핵심 개념

DNS는 도메인 이름을 IP 주소로 변환하고 메일 서버 정보를 찾는 이름 해석 서비스이다. HTTP는 웹 브라우저와 웹 서버 사이에서 웹 문서를 주고받는 프로토콜로 웹메일 접속에 사용된다. TCP는 신뢰성 있는 연결형 전송을 제공하며 SMTP·POP3·IMAP 같은 이메일 프로토콜의 기반이 된다. RTP는 음성·영상과 같은 실시간 데이터를 전송하는 프로토콜이다.

선지별 해설

① O — DNS는 수신 도메인의 메일 서버를 찾을 때 MX 레코드를 조회하므로 이메일 서비스에 사용된다.
② O — HTTP는 브라우저에서 웹메일 화면과 데이터를 주고받을 때 사용된다.
③ X — 이메일 서비스에 사용되는 프로토콜이라는 표현이 틀리다. RTP는 이메일 전송용이 아니라 음성·영상 등 실시간 미디어 데이터를 전송하는 프로토콜이다. 따라서 적절하지 않은 선지이다.
④ O — TCP는 SMTP·POP3·IMAP이 신뢰성 있게 데이터를 주고받도록 하는 전송 계층 프로토콜이므로 이메일 서비스에 사용된다.

오답 포인트

③의 RTP도 네트워크 프로토콜이라는 점만 보고 이메일과 관련 있다고 판단하기 쉽다. 그러나 RTP의 핵심 용도는 실시간 음성·영상 전송이며, 이메일의 송수신이나 웹메일 접속을 담당하지 않는다.

다시 나오면 이렇게 판단

각 프로토콜의 대표 기능을 연결해 판단한다. 도메인·메일 서버 조회는 DNS, 웹메일 접속은 HTTP, 신뢰성 있는 전송 기반은 TCP, 실시간 음성·영상 전송은 RTP이다.

암기 포인트

이메일은 DNS·HTTP·TCP와 관련되고, RTP는 실시간 음성·영상 전송과 관련된다.

NoteBase

문 6. 운영체제 유형에 대한 <보기>의 설명 중 옳은 것의 총 개수는?

<보 기>

ㄱ. 다중 프로그래밍 시스템은 CPU가 유휴상태가 될 때, CPU 작업을 필요로 하는 여러 작업 중 한 작업이 CPU를 사용할 수 있도록 한다.

ㄴ. 다중 처리 시스템에서는 CPU 사이의 연결, 상호작업, 역할 분담 등이 고려되어야 한다.

ㄷ. 시분할 시스템은 CPU가 비선점 스케줄링 방식으로 여러 개의 작업을 교대로 수행한다.

ㄹ. 실시간 처리 시스템은 작업 실행에 대한 시간제약 조건이 있으므로 선점 스케줄링 방식을 이용한다.

ㅁ. 다중 프로그래밍 시스템의 목적은 CPU 활용의 극대화에 있으며, 시분할 시스템은 응답시간의 최소화에 목적이 있다.

① 1개 ② 2개 ③ 3개 ④ 4개

정답 4번

출제 포인트

다중 프로그래밍, 다중 처리, 시분할, 실시간 처리의 목적과 스케줄링 방식을 구분하고 옳은 설명의 개수를 판단하는 문제이다.

핵심 개념

다중 프로그래밍은 여러 작업을 주기억장치에 두고 한 작업이 입출력 등으로 CPU를 쓰지 못할 때 다른 작업에 CPU를 할당하여 CPU 이용률을 높이는 방식이다.
다중 처리는 둘 이상의 CPU가 작업을 분담하여 처리하는 방식이므로 CPU 간 연결, 상호작용, 역할 분담이 중요하다.
시분할은 짧은 시간 할당량을 기준으로 CPU를 선점하여 여러 작업을 번갈아 실행함으로써 빠른 응답을 제공하는 방식이다.
실시간 처리는 정해진 시간 안에 결과를 내는 것이 중요하므로 긴급도나 우선순위가 높은 작업이 CPU를 확보할 수 있는 선점형 스케줄링을 주로 이용한다.

NoteBase

보기 판단

ㄱ O: 한 작업이 입출력 대기 등으로 CPU를 사용하지 못하면 CPU가 필요한 다른 작업을 실행하여 CPU의 유휴 시간을 줄인다.
ㄴ O: 다중 처리 시스템은 여러 CPU가 함께 작동하므로 CPU 간 연결 구조, 상호작용, 작업 및 역할 분담을 고려해야 한다.
ㄷ X: 틀린 표현은 '비선점 스케줄링'이다. 시분할 시스템은 시간 할당량이 끝나면 실행 중인 작업에서 CPU를 회수해 다른 작업에 배정하는 선점 스케줄링 방식으로 여러 작업을 교대로 수행한다.
ㄹ O: 실시간 처리 시스템은 시간 제약을 만족해야 하므로 우선순위가 높은 작업이 즉시 실행될 수 있도록 선점 스케줄링을 이용한다.
ㅁ O: 다중 프로그래밍의 주된 목적은 CPU 이용률의 극대화이고, 시분할의 주된 목적은 사용자의 응답시간 최소화이다.
따라서 옳은 것은 ㄱ, ㄴ, ㄹ, ㅁ의 4개이다.

선지별 해설

① X: 옳은 설명은 1개가 아니라 ㄱ, ㄴ, ㄹ, ㅁ의 4개이다. '1개'는 '4개'로 고쳐야 한다.
② X: 옳은 설명은 2개가 아니라 ㄱ, ㄴ, ㄹ, ㅁ의 4개이다. '2개'는 '4개'로 고쳐야 한다.
③ X: 옳은 설명은 3개가 아니라 ㄱ, ㄴ, ㄹ, ㅁ의 4개이다. '3개'는 '4개'로 고쳐야 한다.
④ O: ㄷ만 틀리고 ㄱ, ㄴ, ㄹ, ㅁ이 옳으므로 옳은 설명의 총 개수는 4개이다.

오답 포인트

ㄷ의 '비선점'이 핵심 함정이다. 시분할은 각 작업에 짧은 시간 할당량을 주고 시간이 끝나면 CPU를 회수해야 하므로 선점 방식이다. 또한 다중 프로그래밍은 CPU 이용률, 시분할은 응답시간을 중시한다는 목적의 차이를 혼동하면 ㅁ을 잘못 판단할 수 있다.

다시 나오면 이렇게 판단

운영체제 유형을 '구성·목적·스케줄링'의 세 기준으로 구분한다. 여러 CPU이면 다중 처리, CPU 유휴 감소이면 다중 프로그래밍, 시간 할당량과 빠른 응답이면 시분할, 마감시간 준수이면 실시간 처리로 판단한다.

암기 포인트

다중 프로그래밍은 CPU 이용률, 시분할은 응답시간, 실시간 처리는 시간 제약이며, 시분할의 시간 할당량은 선점을 뜻한다.

NoteBase

문 7. 가상 메모리에 대한 <보기>의 설명 중 옳은 것을 모두 고른 것은?

<보 기>

ㄱ. 인위적 연속성이란 프로세스의 가상주소 공간상의 연속적인 주소가 실제 기억장치에서도 연속성이 보장되어야 함을 의미한다.

ㄴ. 다중 프로그래밍 정도가 높은 경우, 프로세스가 프로그램 수행시간보다 페이지 교환시간에 더 많은 시간을 소요하고 있다면 스레싱(thrashing) 현상이 발생한 것이다.

ㄷ. 프로세스를 실행하는 동안 일부 페이지만 집중적으로 참조하는 경우를 지역성(locality)이라 하며, 배열 순회는 공간 지역성의 예이다.

ㄹ. 프로세스가 자주 참조하는 페이지의 집합을 작업 집합(working set)이라 하며, 작업 직합은 최초 한번 결정되면 그 이후부터는 변하지 않는다.

① ㄱ, ㄴ ② ㄱ, ㄹ ③ ㄴ, ㄷ ④ ㄴ, ㄷ, ㄹ

정답 3번

출제 포인트

가상 메모리의 인위적 연속성, 스레싱, 지역성, 작업 집합의 의미를 구분하는 문제이다.

핵심 개념

인위적 연속성: 프로세스에는 연속된 가상 주소 공간을 제공하되, 그 페이지들이 실제 물리 메모리에서 연속될 필요는 없다는 성질이다.
스레싱: 페이지 부재가 지나치게 자주 발생하여 프로세스 실행보다 페이지 교환에 더 많은 시간을 쓰는 현상이다.
지역성: 일정 시간 동안 주소 공간의 일부를 집중적으로 참조하는 성질이다. 배열의 연속된 원소를 차례로 순회하는 것은 공간 지역성의 예이다.
작업 집합: 최근의 일정한 시간 구간 또는 참조 횟수 구간에서 프로세스가 참조한 페이지들의 집합으로, 실행 단계에 따라 달라진다.

NoteBase

선지별 해설

① X — ㄱ은 틀리고 ㄴ은 옳으며, 옳은 ㄷ이 빠져 있다. ㄱ의 '실제 기억장치에서도 연속성이 보장되어야 한다'가 틀린 표현이다. 실제로는 가상 주소가 연속되어 보여도 해당 페이지들은 물리 메모리의 불연속 위치에 배치될 수 있다. 올바른 조합은 ㄴ, ㄷ이다.
② X — ㄱ과 ㄹ은 모두 틀렸고, 옳은 ㄴ과 ㄷ이 빠져 있다. ㄱ은 물리적 연속성을 요구하지 않으며, ㄹ의 작업 집합은 고정되지 않고 최근 참조 양상에 따라 변한다. 올바른 조합은 ㄴ, ㄷ이다.
③ O — ㄴ은 페이지 교환 시간이 실제 실행 시간보다 커지는 스레싱을 정확히 설명한다. ㄷ은 일부 페이지를 집중적으로 참조하는 지역성과 배열 순회의 공간 지역성을 정확히 설명한다. 따라서 옳은 것은 ㄴ, ㄷ이다.
④ X — ㄴ과 ㄷ은 옳지만 ㄹ이 틀려서 전체 조합은 옳지 않다. ㄹ의 '최초 한 번 결정되면 그 이후부터는 변하지 않는다'가 틀린 표현이며, 작업 집합은 최근의 참조 구간을 기준으로 계속 갱신된다. 올바른 조합은 ㄴ, ㄷ이다.

오답 포인트

ㄱ은 가상 주소의 연속성을 물리 메모리의 연속성으로 잘못 바꾼 함정이다. ㄹ은 작업 집합을 한 번 정해지는 고정 집합으로 표현했지만, 작업 집합은 프로그램의 실행 단계와 최근 참조 구간에 따라 변한다.

다시 나오면 이렇게 판단

가상 메모리에서는 '가상 주소의 연속성'과 '물리 메모리의 배치'를 분리하여 판단한다. 페이지 교환 시간이 실행 시간보다 길면 스레싱, 최근 일부 주소를 집중 참조하면 지역성, 최근 참조 페이지들의 변화하는 집합이면 작업 집합으로 판별한다.

암기 포인트

가상 주소는 연속·물리 배치는 불연속 가능, 스레싱은 교환 과다, 지역성은 참조 집중, 작업 집합은 최근 참조에 따라 변화한다.

NoteBase

문 8. 운영체제 상의 프로세스(process)에 관한 설명으로 옳지 않은 것은?

① 프로세스의 영역 중 스택 영역은 동적 메모리 할당에 활용된다.

② 디스패치(dispatch)는 CPU 스케줄러가 준비 상태의 프로세스 중 하나를 골라 실행 상태로 바꾸는 작업을 말한다.

③ 프로세스 제어 블록(process control block)은 프로세스 식별자, 메모리 관련 정보, 프로세스가 사용했던 중간값을 포함한다.

④ 문맥교환(context switching)은 CPU를 점유하고 있는 프로세스를 CPU에서 내보내고 새로운 프로세스를 받아들이는 작업이다.

정답 1번

출제 포인트

프로세스의 메모리 영역과 운영체제의 프로세스 관리 개념인 디스패치, 프로세스 제어 블록, 문맥교환을 구분하는 문제이다.

핵심 개념

스택 영역: 함수 호출 정보, 매개변수, 지역 변수 등이 저장되며 호출과 반환에 따라 자동으로 할당·해제되는 영역이다.
힙 영역: 실행 중 필요한 크기의 메모리를 동적으로 할당하고 해제하는 데 사용하는 영역이다.
디스패치: 준비 상태에서 선택된 프로세스에 CPU를 배정하여 실행 상태로 전환하는 과정이다.
프로세스 제어 블록(PCB): 프로세스 식별자, 상태, CPU 레지스터 값, 메모리 관리 정보 등 프로세스 관리에 필요한 정보를 저장하는 자료구조이다.
문맥교환: 실행 중인 프로세스의 문맥을 저장하고 다음 프로세스의 문맥을 복원하여 CPU 실행 대상을 바꾸는 작업이다.

NoteBase

선지별 해설

① X
‘스택 영역은 동적 메모리 할당에 활용된다’는 표현이 틀렸다. 스택은 함수 호출 정보와 지역 변수 등을 자동으로 관리하는 데 쓰이며, 동적 메모리 할당에는 힙 영역이 활용된다.

② O
CPU 스케줄러가 준비 상태의 프로세스 가운데 하나를 선택하면 디스패치 과정에서 그 프로세스에 CPU가 배정되어 실행 상태로 전환된다.

③ O
PCB에는 프로세스 식별자, 메모리 관리 정보와 함께 문맥교환 때 저장해야 하는 프로그램 카운터와 CPU 레지스터 값 등의 중간 상태가 포함된다.

④ O
문맥교환은 현재 CPU를 점유한 프로세스의 상태를 저장하여 내보내고, 새로 실행할 프로세스의 저장된 상태를 복원해 받아들이는 작업이다.

오답 포인트

①의 ‘동적’이라는 말 때문에 스택이 실행 중 크기가 변한다는 사실과 동적 메모리 할당을 혼동하기 쉽다. 스택의 할당·해제는 함수 호출에 따라 자동으로 이루어지지만, 프로그래머가 실행 중 요청하는 동적 메모리는 힙에 할당된다.

다시 나오면 이렇게 판단

메모리 영역 문제에서는 함수 호출·지역 변수·자동 관리는 스택, 실행 중 요청하는 동적 할당은 힙으로 구분한다. 프로세스 상태 문제에서는 준비 프로세스 선택과 CPU 배정은 디스패치, 프로세스 정보 저장은 PCB, 실행 대상 교체를 위한 상태 저장·복원은 문맥교환으로 판단한다.

암기 포인트

지역 변수와 함수 호출은 스택, 동적 할당은 힙, 프로세스 정보는 PCB, CPU 교대는 문맥교환이다.

NoteBase

문 9. 조직의 내부나 외부에 분산된 여러 데이터 소스로부터 필요로 하는 데이터를 검색하여 수동 혹은 자동으로 수집하는 과정과 관련된 기술에 해당하지 않는 것은?

① ETL(Extraction, Transformation, Loading)
② 로그 수집기
③ 맵리듀스(MapReduce)
④ 크롤링(crawling)

정답 3번

출제 포인트

데이터를 여러 원천에서 찾아 모으는 수집 기술과, 이미 확보한 데이터를 분산 처리하는 기술을 구별하는 문제이다.

핵심 개념

데이터 수집은 필요한 데이터를 원천에서 확보하여 저장·처리 단계로 넘기는 과정이다. ETL의 추출 단계, 로그 수집기, 크롤링은 데이터 확보와 직접 관련된다. 반면 맵리듀스는 대규모 데이터를 여러 컴퓨터에 나누어 처리하고 결과를 합치는 분산 처리 모델이다.

선지별 해설

① O — ETL은 여러 데이터 원천에서 데이터를 추출하고, 변환한 뒤, 대상 시스템에 적재하는 과정이므로 데이터 수집과 관련된다.
② O — 로그 수집기는 서버·응용 프로그램·장비 등에서 생성되는 로그를 자동으로 모으는 도구이므로 데이터 수집 기술에 해당한다.
③ X — 맵리듀스는 데이터를 원천에서 검색·수집하는 기술이 아니라, 이미 확보한 대규모 데이터를 Map 단계에서 분산 처리하고 Reduce 단계에서 결과를 집계하는 처리 모델이다. 틀린 표현인 '데이터 수집 기술'은 '대규모 데이터의 분산 처리 기술'로 고쳐야 한다.
④ O — 크롤링은 웹 페이지 등을 자동으로 탐색하면서 필요한 정보를 가져오는 방식이므로 외부 데이터의 자동 수집과 관련된다.

오답 포인트

③ 맵리듀스도 빅데이터 분야에서 쓰이므로 수집 기술처럼 보일 수 있지만, 핵심 기능은 수집이 아니라 분할 처리와 결과 집계이다.

NoteBase

다시 나오면 이렇게 판단

기술이 데이터를 원천에서 가져오는지, 가져온 데이터를 계산·가공하는지를 구분한다. 가져오면 수집 기술이고, 분할 계산과 집계를 수행하면 처리 기술이다.

암기 포인트

ETL·로그 수집기·크롤링은 수집, 맵리듀스는 분산 처리이다.

NoteBase

문 10. 기계학습(machine learning)에 대한 설명으로 옳지 않은 것은?

① 강화학습은 기계가 환경과 상호작용하면서 시행착오 과정에서의 보상을 통해 학습을 수행한다.

② 기계학습 모델의 성능 기준으로 사용되는 F1 점수(score)는 정밀도(precision)와 검출률(recall)을 동시에 고려한 조화평균 값이다.

③ 치매환자의 뇌 영상 분류를 위해서 기존에 잘 만들어진 영상 분류 모델에 새로운 종류의 뇌 영상 데이터를 확장하여 학습시키는 방법은 전이학습(transfer learning)의 예이다.

④ 비지도학습은 라벨(label) 정보를 포함하고 있는 훈련 데이터를 사용하며, 주가나 환율 변화, 유가 예측 등의 회귀(regression) 문제에 적용된다.

정답 4번

출제 포인트

강화학습, F1 점수, 전이학습, 지도학습과 비지도학습의 핵심 특징을 구별하는 문제이다.

핵심 개념

강화학습: 에이전트가 환경과 상호작용하며 행동의 결과로 받은 보상을 바탕으로 더 나은 행동 전략을 학습하는 방식이다. / 키워드: 환경·행동·보상

F1 점수: 정밀도와 재현율의 조화평균으로, 두 지표를 함께 고려하는 평가 지표이다. / 키워드: 정밀도·재현율·조화평균

전이학습: 한 데이터나 과제로 미리 학습한 모델의 지식을 새로운 관련 데이터나 과제에 활용하는 방식이다. / 키워드: 사전학습 모델·지식 재사용

지도학습과 비지도학습: 지도학습은 입력과 정답 라벨이 함께 있는 데이터로 학습하고, 비지도학습은 정답 라벨이 없는 데이터에서 구조나 패턴을 찾는다. / 결정 차이축: 정답 라벨의 유무

NoteBase

선지별 해설

① O — 강화학습은 기계가 환경과 상호작용하면서 시행착오를 겪고, 그 과정에서 얻는 보상을 기준으로 행동 전략을 학습하므로 옳다.
② O — F1 점수는 정밀도와 재현율을 동시에 반영하는 조화평균이므로 옳다. 한 지표만 높고 다른 지표가 낮은 경우를 균형 있게 평가한다.
③ O — 기존 영상 분류 모델이 학습한 특징을 새로운 종류의 뇌 영상 데이터 학습에 재사용하는 것은 전이학습의 사례이므로 옳다.
④ X — 틀린 표현은 '비지도학습은 라벨 정보를 포함하고 있는 훈련 데이터를 사용'한다는 부분이다. 올바르게는 지도학습이 라벨이 있는 훈련 데이터를 사용하고, 비지도학습은 라벨이 없는 데이터의 구조나 패턴을 학습한다. 주가·환율·유가처럼 연속적인 수치를 예측하는 회귀는 정답 수치가 있는 자료로 학습할 때 지도학습에 해당한다.

오답 포인트

④는 회귀 문제의 예시 자체보다 '비지도학습'과 '라벨 정보를 포함한 훈련 데이터'를 연결한 부분이 함정이다. 라벨이 있는 데이터는 지도학습의 기준이며, 회귀도 보통 지도학습의 대표 과제이다.

다시 나오면 이렇게 판단

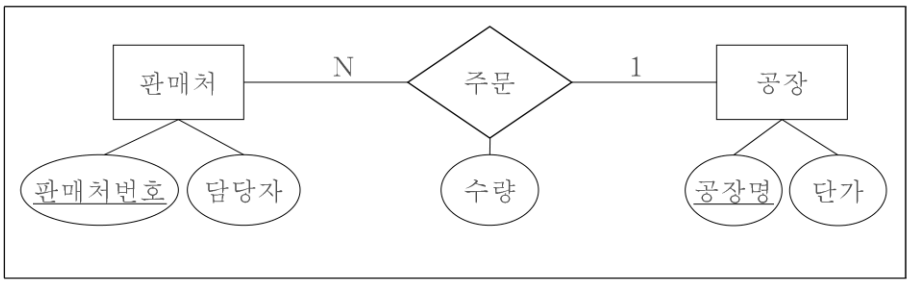
학습 방식은 핵심 신호로 구별한다. 보상을 받으며 행동을 개선하면 강화학습, 라벨이 있으면 지도학습, 라벨 없이 패턴을 찾으면 비지도학습, 기존 모델의 지식을 새 과제에 활용하면 전이학습이다.

암기 포인트

보상은 강화학습, 라벨은 지도학습, 무라벨은 비지도학습, 기존 지식 재사용은 전이학습이다. F1은 정밀도와 재현율의 조화평균이다.

NoteBase

문 11. 다음 E-R 다이어그램을 관계형 스키마로 올바르게 변환한 것은? (단, 속성명의 밑줄은 해당 속성이 기본키임을 의미한다.)



① 판매처(판매처번호, 담당자)
공장(공장명, 단가, 판매처번호, 수량)

② 판매처(판매처번호, 담당자, 공장명, 수량)
공장(공장명, 단가)

③ 판매처(판매처번호, 담당자)
주문(판매처번호, 수량)
공장(공장명, 단가)

④ 판매처(판매처번호, 담당자)
주문(공장명, 수량)
공장(공장명, 단가)

정답 2번

출제 포인트

E-R 다이어그램의 1:N 관계를 릴레이션으로 변환하는 규칙을 묻는 문제이다. 그림에서 판매처는 N측, 공장은 1측이며, 주문의 속성은 수량이다. 1측인 공장의 기본키 공장명과 관계 속성 수량을 N측인 판매처 릴레이션에 포함한다.

NoteBase

선지별 해설

① X
공장 릴레이션에 판매처번호와 수량을 넣으면 한 공장과 연결되는 여러 판매처를 한 행으로 표현할 수 없다. 틀린 표현인 '공장(공장명, 단가, 판매처번호, 수량)'은 '판매처(판매처번호, 담당자, 공장명, 수량), 공장(공장명, 단가)'로 고쳐야 한다.

② O
N측인 판매처에 1측 공장의 기본키 공장명을 외래키로 넣고 관계 속성 수량도 함께 넣었다. 따라서 판매처(판매처번호, 담당자, 공장명, 수량)와 공장(공장명, 단가)로 변환한 것이 옳다.

③ X
별도의 주문 릴레이션을 만들면서 공장명이 빠져 있어 어느 공장과의 주문 관계인지 나타낼 수 없다. 틀린 표현인 '주문(판매처번호, 수량)'은 별도 릴레이션을 사용한다면 '주문(판매처번호, 공장명, 수량)'으로 고쳐야 하며, 이 문제의 표준적인 1:N 변환에서는 주문 릴레이션을 따로 만들지 않고 공장명과 수량을 판매처에 포함한다.

④ X
공장명은 1측의 기본키이므로 여러 판매처가 같은 공장을 주문하는 상황에서 주문 릴레이션의 기본키가 될 수 없고, 판매처번호도 없어 주문 주체를 식별할 수 없다. 틀린 표현인 '주문(공장명, 수량)'은 '판매처(판매처번호, 담당자, 공장명, 수량)'로 고치고 주문 릴레이션은 따로 만들지 않는다.

오답 포인트

그림의 N과 1을 반대로 읽어 ①처럼 N측 기본키를 1측에 넣는 것이 핵심 함정이다. 1:N 관계에서는 여러 개가 존재하는 N측 릴레이션에 1측 기본키가 들어가야 하며, 관계 자체의 속성인 수량도 같은 N측에 들어간다.

다시 나오면 이렇게 판단

먼저 관계선의 1측과 N측을 찾는다. 그다음 1측의 기본키를 N측에 외래키로 옮기고, 관계의 속성도 N측에 포함했는지 확인한다.

암기 포인트

1:N 변환은 '1측 기본키와 관계 속성을 N측으로' 정리된다.

NoteBase

문 12. 데이터베이스 상의 병행제어를 위한 로킹(locking) 기법에 대한 <보기>의 설명 중 옳은 것의 총 개수는?

<보 기>

ㄱ. 로크(lock)는 하나의 트랜잭션이 데이터를 접근하는 동안 다른 트랜잭션이 그 데이터를 접근할 수 없도록 제어하는 데 쓰인다.
ㄴ. 트랜잭션이 로크한 데이터에 대해서는 해당 트랜잭션이 종료되기 전에 해당 데이터에 대한 언로크(unlock)를 실행하여야 한다.
ㄷ. 로킹의 단위가 작아질수록 로크의 수가 많아서 관리가 복잡해지지만 병행성 수준은 높아지는 장점이 있다.
ㄹ. 2단계 로킹 규약을 적용하면 트랜잭션의 직렬 가능성을 보장할 수 있어서 교착상태 발생을 예방할 수 있다.

① 1개 ② 2개 ③ 3개 ④ 4개

정답 3번

출제 포인트

로크의 목적, 로킹 단위와 병행성의 관계, 2단계 로킹의 효과를 각각 판단하여 옳은 설명의 개수를 세는 문제이다.

핵심 개념

로크는 한 트랜잭션이 사용하는 데이터에 다른 트랜잭션이 충돌하는 방식으로 접근하지 못하도록 제어하는 장치이다. 로킹 단위가 작아지면 관리할 로크 수와 관리 부담은 커지지만 서로 다른 작은 단위를 동시에 사용할 수 있어 병행성은 높아진다. 2단계 로킹은 로크를 획득하는 확장 단계와 로크를 해제하는 축소 단계를 구분하여 직렬 가능성을 보장하지만, 트랜잭션들이 서로의 로크 해제를 기다리는 교착상태까지 예방하지는 못한다. 따라서 ㄱ, ㄴ, ㄷ은 옳고 ㄹ은 틀려 옳은 설명은 모두 3개이다.

NoteBase

선지별 해설

① X — 옳은 설명은 1개가 아니라 ㄱ, ㄴ, ㄷ의 3개이다. 틀린 표현은 '1개'이며, 올바르게는 '3개'이다.
② X — 옳은 설명은 2개가 아니라 3개이다. ㄱ은 로크의 목적, ㄴ은 종료 전 언로크, ㄷ은 작은 로킹 단위의 높은 병행성을 바르게 설명한다. 틀린 표현은 '2개'이며, 올바르게는 '3개'이다.
③ O — ㄱ, ㄴ, ㄷ이 옳고 ㄹ이 틀리므로 옳은 설명의 총개수는 3개이다.
④ X — ㄹ까지 옳다고 보아 4개로 세면 안 된다. ㄹ의 틀린 표현은 '교착상태 발생을 예방할 수 있다'이며, 올바르게는 '2단계 로킹은 직렬 가능성을 보장하지만 교착상태 발생을 예방하지는 못한다'이다. 따라서 '4개'가 아니라 '3개'이다.

오답 포인트

ㄹ에서 '직렬 가능성 보장'과 '교착상태 예방'을 같은 효과로 묶은 것이 핵심 함정이다. 2단계 로킹 규약을 지켜도 두 트랜잭션이 서로 상대방의 로크 해제를 기다리는 순환 대기가 생길 수 있다.

다시 나오면 이렇게 판단

각 설명을 먼저 독립적으로 O/X 판단한 뒤 O의 개수를 센다. 로킹 단위가 작아지면 로크 수·관리 부담·병행성은 증가하고, 2단계 로킹은 직렬 가능성을 보장하지만 교착상태를 자동으로 막지는 않는다는 기준을 적용한다.

암기 포인트

작은 로킹 단위는 로크 수와 병행성이 함께 증가한다. 2단계 로킹은 직렬 가능성 O, 교착상태 예방 X이다.

NoteBase

문 13. 기능점수에 대한 <보기>의 설명 중 옳은 것의 총 개수는?

<보 기>

ㄱ. 소프트웨어가 사용자에게 제공하는 기능의 수를 수치로 정량화하여 소프트웨어의 규모를 산정하는 데 주로 사용한다.
ㄴ. 트랜잭션의 기능을 측정하기 위한 기준으로 내부입력, 내부출력, 내부조회가 있다.
ㄷ. 응용 패키지의 규모 산정, 소프트웨어의 품질 및 생산성 분석, 소프트웨어 개발과 유지보수를 위한 비용 및 소요자원 산정 등에 사용할 수 있다.
ㄹ. 기능점수 산출 시 적용되는 조정인자는 시스템의 특성을 반영하지 않는다.

① 1개 ② 2개 ③ 3개 ④ 4개

정답 2번

출제 포인트

기능점수의 의미, 트랜잭션 기능의 유형, 활용 분야, 조정인자의 역할을 구분하여 옳은 보기의 수를 판단하는 문제이다.

핵심 개념

기능점수는 소프트웨어가 사용자에게 제공하는 기능의 양을 정량화하여 소프트웨어 규모를 측정하는 방법이다. 기능점수 산정에서는 데이터 기능과 트랜잭션 기능을 식별하며, 트랜잭션 기능에는 외부입력, 외부출력, 외부조회가 포함된다. 조정인자는 시스템의 일반적 특성을 반영하여 기능점수를 보정하는 요소이다.

보기 판단

ㄱ O: 사용자가 인식하는 기능의 수를 정량화하여 소프트웨어 규모를 산정하는 것이 기능점수의 기본 목적이다.
ㄴ X: 틀린 표현은 '내부입력, 내부출력, 내부조회'이다. 트랜잭션 기능의 유형은 외부입력, 외부출력, 외부조회이다.
ㄷ O: 기능점수는 응용 패키지의 규모 산정, 품질·생산성 분석, 개발·유지보수 비용 및 자원 산정 등에 활용할 수 있다.
ㄹ X: 틀린 표현은 '시스템의 특성을 반영하지 않는다'이다. 실제로 조정인자는 시스템의 특성을 반영하여 기능점수를 보정한다. 따라서 옳은 것은 ㄱ과 ㄷ의 2개이다.

NoteBase

선지별 해설

① X: 옳은 보기는 ㄱ과 ㄷ이므로 총 1개가 아니라 2개이다. 올바르게는 '2개'이다.
② O: 옳은 보기는 ㄱ과 ㄷ의 총 2개이므로 정답이다.
③ X: ㄴ과 ㄹ은 틀리므로 옳은 보기는 3개가 아니라 2개이다. 올바르게는 '2개'이다.
④ X: 네 보기 중 ㄴ과 ㄹ이 틀리므로 옳은 보기는 4개가 아니라 2개이다. 올바르게는 '2개'이다.

오답 포인트

ㄴ의 '내부'라는 표현과 ㄹ의 '반영하지 않는다'라는 부정 표현이 함정이다. 트랜잭션 기능은 외부입력·외부출력·외부조회이고, 조정인자는 시스템 특성을 반영한다.

다시 나오면 이렇게 판단

기능점수 문항에서는 먼저 사용자 기능의 정량화라는 목적을 확인하고, 트랜잭션 기능의 '외부'라는 용어와 조정인자의 '시스템 특성 반영' 여부를 기준으로 각 보기를 판별한 뒤 O의 개수를 센다.

암기 포인트

기능점수는 사용자 기능으로 규모를 재고, 트랜잭션은 외부입력·외부출력·외부조회이며, 조정인자는 시스템 특성을 반영한다.

NoteBase

문 14. 소프트웨어 테스트에 대한 설명으로 옳지 않은 것은?

① 통합 테스트는 단위 테스트가 끝난 모듈들을 통합하여 모듈 간의 인터페이스 관련 오류가 있는지를 찾는 검사이다.

② 테스트의 목적은 소프트웨어 요구사항의 만족도 및 예상 결과와 실제 결과의 차이점을 파악함으로써 소프트웨어의 오류를 찾아내는 것이다.

③ 화이트 박스 테스트는 프로그램 원시 코드의 논리적 구조를 체계적으로 점검하며, 프로그램 구조에 의거하여 검사한다.

④ 블랙 박스 테스트에는 기초 경로(basic path), 조건 기준(condition coverage), 루프(loop) 검사, 논리 위주(logic driven) 검사 등이 있다.

정답 4번

출제 포인트

소프트웨어 테스트의 목적과 단계별 테스트의 특징을 이해하고, 프로그램 내부 구조를 기준으로 하는 화이트박스 테스트와 입출력 기능을 기준으로 하는 블랙박스 테스트를 구분하는 문제이다.

핵심 개념

통합 테스트: 단위 테스트가 끝난 모듈들을 결합하여 모듈 사이의 인터페이스와 상호 작용에서 발생하는 오류를 찾는 테스트이다.
화이트박스 테스트: 프로그램의 내부 논리와 제어 구조를 바탕으로 경로, 조건, 반복 구조 등을 점검하는 테스트이다. 기초 경로 검사, 조건 검사, 루프 검사 등이 대표적이다.
블랙박스 테스트: 프로그램의 내부 구조를 보지 않고 요구 명세에 따른 입력과 출력의 관계 및 기능을 검사하는 테스트이다. 동등 분할, 경계값 분석, 원인-결과 그래프 등이 대표적이다.
화이트박스는 내부 구조 중심이고, 블랙박스는 외부 기능과 입출력 중심이다.

NoteBase

선지별 해설

① O — 통합 테스트는 단위 테스트를 마친 모듈들을 결합한 뒤 모듈 간 인터페이스와 상호 작용의 오류를 찾는 검사이므로 옳다.
② O — 소프트웨어 테스트는 요구사항의 충족 여부를 확인하고 예상 결과와 실제 결과의 차이를 통해 결함을 발견하는 활동이므로 옳다.
③ O — 화이트박스 테스트는 프로그램 원시 코드의 내부 논리와 제어 구조를 체계적으로 살펴보는 구조 기반 검사이므로 옳다.
④ X — 틀린 표현은 '블랙 박스 테스트에는 기초 경로, 조건 기준, 루프 검사, 논리 위주 검사 등이 있다'이다. 이러한 기법은 프로그램 내부 논리 구조를 살피는 화이트박스 테스트에 해당한다. 올바르게는 '화이트박스 테스트에는 기초 경로 검사, 조건 검사, 루프 검사 등 구조 기반 검사 기법이 있다'이다. 블랙박스 테스트의 대표 기법에는 동등 분할, 경계값 분석, 원인-결과 그래프 등이 있다.

오답 포인트

④는 검사 기법의 이름 자체는 실제로 존재하지만 소속을 블랙박스로 바꾸어 놓았다. '경로·조건·루프·논리'는 코드 내부 구조를 직접 확인한다는 신호이므로 화이트박스에 해당한다.

다시 나오면 이렇게 판단

선지에 경로, 조건, 분기, 루프처럼 코드 내부 제어 구조를 나타내는 말이 나오면 화이트박스로 판단한다.
입력값의 구간, 경계, 명세상의 기능과 출력 결과를 중심으로 검사하면 블랙박스로 판단한다.

암기 포인트

화이트박스는 코드 안의 경로·조건·루프, 블랙박스는 코드 밖의 입력·출력·기능이다.

NoteBase

문 15. 컴퓨터 메모리 용량이 8K×32Bit라 하면, MAR(Memory Address Register)과 MBR(Memory Buffer Register)은 각각 몇 비트인가?

① MAR: 8 MBR: 32
② MAR: 32 MBR: 8
③ MAR: 13 MBR: 8
④ MAR: 13 MBR: 32

정답 4번

출제 포인트

메모리의 워드 수로 MAR의 비트 수를, 한 워드의 길이로 MBR의 비트 수를 구하는 문제이다.

조건 정리

주어진 값: 메모리 용량은 $8K \times 32bit$이다.
구해야 할 값: MAR과 MBR의 비트 수이다.
주의할 조건: $1K = 2^{10}$이므로 $8K = 8 \times 2^{10} = 2^{13}$개의 워드이며, 각 워드의 길이는 $32bit$이다.

사용할 공식

MAR 비트 수=_2(주소 지정 가능한 워드 수)
MBR 비트 수=한 워드의 비트 수
MAR은 접근할 메모리 주소를 저장하므로 전체 워드 수를 구별할 만큼의 비트가 필요하고, MBR은 한 번에 읽거나 쓰는 한 워드를 저장하므로 워드 길이와 같은 비트 수가 필요하다.

계산 과정

1. $8K = 8 \times 2^{10} = 2^{13}$이므로 주소는 2^{13}개이다.
2. 2^{13}개의 주소를 구별하려면 MAR은 $13bit$가 필요하다.
3. 한 워드가 $32bit$이므로 MBR은 $32bit$이다.

NoteBase

선지 판단

① X — ‘MAR: 8’이 틀렸다. 8은 8*K*의 계수일 뿐 주소 비트 수가 아니다. 올바르게는 MAR: 13, MBR: 32이다.
② X — ‘MAR: 32, MBR: 8’이 모두 틀렸다. 32는 워드 길이이므로 MBR에 해당하며, 주소 수 $8K = 2^{13}$에서 MAR은 13비트이다. 올바르게는 MAR: 13, MBR: 32이다.
③ X — ‘MBR: 8’이 틀렸다. MBR은 한 워드 전체를 저장하므로 32비트이다. 올바르게는 MAR: 13, MBR: 32이다.
④ O — $8K = 2^{13}$개의 주소에는 13비트 MAR이 필요하고, 한 워드가 32비트이므로 MBR은 32비트이다. 계산 결과와 모두 일치한다.

오답 포인트

‘8’, ‘13’, ‘32’가 각각 무엇을 뜻하는지 혼동하는 것이 함정이다. 8을 그대로 MAR 비트 수로 쓰거나, 워드 길이 32를 MAR에 배치하면 틀린다. 8*K*는 먼저 2^{13}으로 바꾸어 주소 비트 수를 구하고, 32비트는 그대로 MBR의 크기로 판단한다.

다시 나오면 이렇게 판단

메모리가 ‘워드 수 × 워드 길이’로 주어지면, 워드 수를 2^n으로 바꾸어 MAR을 n비트로 정하고 워드 길이를 MBR의 비트 수로 정한다.

암기 포인트

MAR은 주소의 개수에 로그를 취하고, MBR은 한 워드의 길이를 그대로 쓴다.

NoteBase

문 16. RAID(Redundant Array of Inexpensive Disks)에 대한 설명으로 옳지 않은 것은?

① RAID 1은 디스크 미러링(disk mirroring) 방식으로, 디스크 오류 시 데이터 복구가 가능하지만 디스크 용량의 효율성이 떨어진다.

② RAID 3은 데이터를 비트 또는 바이트 단위로 여러 디스크에 분할 저장하는 방식으로, 디스크 접근 속도가 향상되지는 않지만 쓰기 동작 시 시간 지연이 발생하지 않는다.

③ RAID 4는 데이터를 블록 단위로 여러 디스크에 분할 저장하는 방식으로, 오류의 검출 및 정정을 위해 별도의 패리티 비트를 사용한다.

④ RAID 5는 패리티 블록들을 여러 디스크에 분산 저장하는 방식으로, 단일 오류 검출 및 정정이 가능하다.

정답 2번

출제 포인트

RAID 1·3·4·5의 데이터 분산 단위, 패리티 배치 방식, 성능과 장애 복구 특성을 구별하는 문제이다.

핵심 개념

RAID는 여러 디스크를 묶어 성능이나 신뢰성을 높이는 저장 방식이다.
RAID 1은 같은 데이터를 두 디스크에 복제하는 미러링 방식이다.
RAID 3은 비트 또는 바이트 단위로 데이터를 분산하고 전용 패리티 디스크를 사용한다.
RAID 4는 블록 단위로 데이터를 분산하고 전용 패리티 디스크를 사용한다.
RAID 5는 블록 단위로 데이터를 분산하면서 패리티도 여러 디스크에 분산한다.

NoteBase

선지별 해설

① O — RAID 1은 동일한 데이터를 복제하는 디스크 미러링 방식이므로 한 디스크에 오류가 생겨도 다른 디스크로 데이터를 복구할 수 있다. 같은 데이터를 중복 저장하므로 실제 사용 가능한 용량의 효율은 낮아진다.
② X — RAID 3은 비트 또는 바이트 단위로 데이터를 여러 디스크에 분산 저장한다. 여러 디스크가 병렬로 동작하므로 대용량 연속 데이터의 전송률은 높일 수 있지만, 디스크 접근 시간 자체가 향상되는 것은 아니다. 따라서 '디스크 접근 속도가 향상되지 않는다'는 부분은 전송률과 구별하면 옳다. 틀린 부분은 '쓰기 동작 시 시간 지연이 발생하지 않는다'이다. 쓰기 때에는 전용 패리티 디스크의 패리티를 함께 처리해야 하므로 지연과 병목이 발생할 수 있다.
③ O — RAID 4는 데이터를 블록 단위로 여러 디스크에 분산하고, 별도의 전용 패리티 디스크에 오류 검출과 복구에 필요한 패리티를 저장한다.
④ O — RAID 5는 블록 단위로 데이터를 분산하며 패리티 블록도 여러 디스크에 나누어 저장한다. 이 패리티를 이용해 하나의 디스크에 장애가 발생한 경우 데이터를 복구할 수 있다.

오답 포인트

②는 RAID 3의 데이터 분산 단위와 디스크 접근 시간에 관한 설명은 맞지만, 쓰기 지연이 발생하지 않는다고 한 부분이 틀렸다. RAID 3은 대용량 연속 데이터의 전송률을 높일 수 있으나, 전용 패리티 처리로 인해 쓰기 지연이나 병목이 발생할 수 있다.

다시 나오면 이렇게 판단

각 RAID를 '분산 단위-패리티 위치-장애 복구-쓰기 부담'의 순서로 판단한다. RAID 3은 비트·바이트 단위와 전용 패리티, RAID 4는 블록 단위와 전용 패리티, RAID 5는 블록 단위와 분산 패리티로 구별한다.

암기 포인트

1은 미러링, 3은 비트·바이트+전용 패리티, 4는 블록+전용 패리티, 5는 블록+분산 패리티이다.

NoteBase

문 17. 다음 워크시트의 [A6]셀과 [A7]셀에 아래와 같이 입력하였다. [A6]과 [A7]의 결과값을 순서대로 바르게 나타낸 것은?

```
[A6] 셀 : =HLOOKUP(11, B1:D5, 3)
[A7] 셀 : =VLOOKUP("나", A2:D5, 4, TRUE)
```

	A	B	C	D
1		10	20	30
2	가	10원	50원	90원
3	나	20원	60원	100원
4	다	30원	70원	110원
5	라	40원	80원	120원

① 20원, 100원 ② 20원, 120원
③ 60원, 100원 ④ 60원, 120원

정답 1번

출제 포인트

이 문제는 HLOOKUP과 VLOOKUP의 검색 방향, 근삿값 검색 방식, 행·열 번호의 기준을 이용해 두 셀의 결과값을 구하는 문제이다.

조건 정리

주어진 값: A6에는 HLOOKUP(11, B1:D5, 3), A7에는 VLOOKUP("나", A2:D5, 4, TRUE)가 입력되어 있다.
구해야 할 값: A6과 A7의 결과값이다.
주의할 조건: HLOOKUP의 네 번째 인수를 생략하면 TRUE가 적용되어 근삿값을 찾는다. 행 번호와 열 번호는 지정한 범위의 첫 행과 첫 열부터 센다.

사용할 공식

HLOOKUP은 범위의 첫 행에서 검색값을 찾고, 지정된 행 번호에 있는 값을 반환한다. TRUE 방식에서는 검색값 이하인 값 중 가장 큰 값을 선택한다.
VLOOKUP은 범위의 첫 열에서 검색값을 찾고, 지정된 열 번호에 있는 값을 반환한다.
이 원리를 쓰는 이유: A6은 가로 방향으로 11에 대응하는 열을 찾아야 하고, A7은 세로 방향으로 '나'에 대응하는 행을 찾아야 하기 때문이다.

NoteBase

계산 과정

1. A6의 검색 범위 B1:D5의 첫 행은 10, 20, 30이다. 11을 TRUE 방식으로 검색하면 11 이하의 최댓값인 10이 있는 B열을 선택한다.
2. 범위 B1:D5의 세 번째 행은 시트의 3행이므로 B3의 값 20원을 반환한다. 따라서 A6은 20원이다.
3. A7은 범위 A2:D5의 첫 열에서 '나'를 찾는다. '나'가 있는 행의 네 번째 열은 D3이고, 그 값은 100원이다. 따라서 A7은 100원이다.

선지 판단

① O — A6은 20원이고 A7은 100원이므로 두 결과를 순서대로 정확히 나타낸다.
② X — A6의 20원은 맞지만 A7의 120원이 틀리다. '나'가 있는 행의 네 번째 열은 D3이므로 올바르게는 20원, 100원이다.
③ X — A7의 100원은 맞지만 A6의 60원이 틀리다. 행 번호 3은 시트의 열 C를 고르는 번호가 아니라 범위 B1:D5에서 반환할 세 번째 행을 뜻하므로 올바르게는 20원, 100원이다.
④ X — A6의 60원과 A7의 120원이 모두 틀리다. A6은 B3의 20원, A7은 D3의 100원을 반환하므로 올바르게는 20원, 100원이다.

오답 포인트

HLOOKUP의 행 번호 3을 열 위치로 오해하면 60원을 고르기 쉽다. 또한 VLOOKUP의 열 번호 4를 시트 전체의 행이나 마지막 데이터 행으로 오해하면 120원을 고르기 쉽다. 행·열 번호는 반드시 지정된 범위 안에서 센다.

다시 나오면 이렇게 판단

먼저 HLOOKUP은 첫 행을 가로로, VLOOKUP은 첫 열을 세로로 검색한다는 점을 구분한다. 그다음 검색 방식이 TRUE인지 FALSE인지 확인하고, 마지막으로 반환 행·열 번호를 지정 범위의 시작점부터 세어 결과 셀을 찾는다.

암기 포인트

HLOOKUP은 가로의 첫 행, VLOOKUP은 세로의 첫 열을 검색한다. TRUE는 근삿값 검색이며, 반환 번호는 지정 범위 내부를 기준으로 센다.

NoteBase

문 18. 프로그래밍 언어 번역 프로그램에 대한 설명으로 옳지 않은 것은?

① 인터프리터(interpreter)는 고급언어로 작성된 원시 프로그램을 함수 단위로 읽어 기계어로 번역하는 프로그램이다.

② 컴파일러(compiler)는 고급언어로 작성된 원시 프로그램을 기계어나 어셈블리어로 된 목적 프로그램으로 바꾸는 프로그램이다.

③ 어셈블러(assembler)는 어셈블리어로 작성된 원시 프로그램을 기계어로 번역하는 프로그램이다.

④ 프리프로세서(preprocessor)는 컴파일러가 컴파일을 수행하기 전에 원시 프로그램의 내용을 변경하는 것이다.

정답 1번

출제 포인트

프로그래밍 언어 번역 프로그램인 인터프리터, 컴파일러, 어셈블러, 프리프로세서의 입력 언어, 출력 언어, 처리 시점을 구별하는 문제이다.

핵심 개념

인터프리터: 고급언어로 작성된 원시 프로그램을 명령문 단위로 해석하여 곧바로 실행하는 번역 프로그램이다.
컴파일러: 고급언어 원시 프로그램 전체를 기계어나 어셈블리어 형태의 목적 프로그램으로 번역한다.
어셈블러: 어셈블리어 원시 프로그램을 기계어 목적 프로그램으로 번역한다.
프리프로세서: 컴파일 전에 매크로 치환, 파일 포함, 조건부 컴파일 등의 전처리를 수행하여 원시 프로그램의 내용을 변환한다.

선지별 해설

① X — 틀린 표현은 '함수 단위로 읽어'이다. 인터프리터는 일반적으로 원시 프로그램을 명령문 단위로 해석하고 즉시 실행하는 프로그램이다. 따라서 올바르게는 '고급언어로 작성된 원시 프로그램을 명령문 단위로 읽어 기계어로 번역·실행하는 프로그램'이다.
② O — 컴파일러는 고급언어로 작성된 원시 프로그램을 기계어나 어셈블리어 형태의 목적 프로그램으로 바꾸는 번역 프로그램이다.
③ O — 어셈블러는 어셈블리어로 작성된 원시 프로그램을 기계어 목적 프로그램으로 번역한다.
④ O — 프리프로세서는 컴파일러가 본격적으로 컴파일하기 전에 원시 프로그램에 매크로 치환, 파일 포함 등의 전처리를 수행한다.

NoteBase

오답 포인트

①의 '함수 단위'가 함정이다. 인터프리터와 컴파일러의 핵심 차이는 함수의 유무가 아니라, 인터프리터는 명령문 단위로 해석·실행하고 컴파일러는 원시 프로그램 전체를 번역하여 목적 프로그램을 만든다는 점이다.

다시 나오면 이렇게 판단

각 번역 프로그램의 입력과 출력을 연결해 판단한다. 고급언어를 명령문 단위로 해석·실행하면 인터프리터, 고급언어 전체를 목적 프로그램으로 번역하면 컴파일러, 어셈블리어를 기계어로 번역하면 어셈블러, 컴파일 전에 원시 프로그램을 변환하면 프리프로세서이다.

암기 포인트

인터프리터는 명령문별 해석·실행, 컴파일러는 전체 번역, 어셈블러는 어셈블리어에서 기계어로 번역, 프리프로세서는 컴파일 전 처리이다.

NoteBase

문 19. 다음 글에서 밑줄 친 부분이 문법상 옳지 않은 것은?

The major source of anger is frustration. If one wants to go somewhere, perform some act, or ① obtain something and is prevented, we say that person is frustrated. One of the basic tenets ② is that frustration tends to arouse aggressive feelings. The behavioral effects of frustration were demonstrated in a classic study. Children were shown a room full of attractive toys but were not allowed to enter it. They stood outside looking at the toys, wanting to play with them, but were unable to reach them. After they had waited for some time, they were allowed to play with them. Other children were given the toys without first ③ being prevented from playing with them. The children who had been frustrated smashed the toys on the floor, threw them against the wall, and generally behaved very ④ destructive.

정답 4번

출제 포인트

병렬 구조, 주어와 동사의 수 일치, 전치사 뒤 동명사, 형용사와 부사의 쓰임을 구별하는 문제이다.

핵심 개념

병렬 구조: 등위접속사로 연결된 요소는 같은 문법 형태를 취한다.
수 일치: one of + 복수 명사에서 문장의 중심 주어는 one이므로 단수 동사를 쓴다.
전치사와 동명사: 전치사 뒤에는 동명사를 쓰며, 수동의 뜻이면 being + 과거분사를 쓴다.
형용사와 부사: 형용사는 주로 명사를 꾸미거나 보어가 되고, 일반동사의 행동 방식은 부사가 꾸민다.

NoteBase

선지별 해설

① O — wants to 뒤에서 go, perform, obtain이 모두 동사원형으로 병렬을 이루므로 문법상 옳다.
② O — One of the basic tenets에서 중심 주어는 단수인 One이므로 단수 동사 is가 옳다.
③ O — 전치사 without 뒤에 동명사 being이 왔고, 아이들이 놀이를 방해받는 수동 의미이므로 being prevented가 옳다. 또한 prevent A from -ing의 구조에 따라 from playing도 적절하다.
④ X — 틀린 표현은 behaved very destructive이다. 일반동사 behaved의 행동 방식을 꾸며야 하므로 형용사 destructive가 아니라 부사 destructively를 써서 behaved very destructively로 고쳐야 한다.

오답 포인트

④의 destructive는 의미상 자연스러워 보이지만, 동사 behaved를 수식하는 자리이므로 품사가 맞지 않는다. 의미만 보지 않고 수식 대상이 동사인지 명사인지 확인해야 한다.

다시 나오면 이렇게 판단

밑줄 친 말이 무엇을 수식하는지 먼저 확인한다. 일반동사의 동작 방식이나 정도를 꾸미면 부사를 쓰고, 명사를 꾸미거나 보어 자리에 놓이면 형용사를 쓴다. 병렬 표현은 연결된 항목들의 형태가 같은지도 함께 확인한다.

암기 포인트

행동을 나타내는 동사를 꾸미는 말은 부사이다: behave destructively.

NoteBase

문 20. 다음 글의 빈칸 (A), (B)에 들어갈 말로 가장 적절한 것은?

Related to the question of revealing persuasive intent is the question of whether a speaker should state conclusions clearly or leave them implied for the audience to discover. Intuitively, we recognize that individuals may more readily embrace their own conclusions than they do those offered by others. For example, psychiatrists prefer to let their patients discover the causes of their mental condition for themselves rather than tell them _____(A)_____. Consequently, speakers may think it wise to merely imply their claims and let listeners draw their own conclusions, especially when source credibility is not high. Such a strategy is dangerous, however, particularly if the audience lacks intelligence or is highly opinionated, because they may draw an incorrect conclusion or distort the speaker's point. The safer approach is to state conclusions _____(B)_____.

(A)	(B)
① directly	explicitly
② directly	implicitly
③ indirectly	explicitly
④ indirectly	implicitly

정답 1번

출제 포인트

이 문제는 지문의 대조 관계를 바탕으로 '직접 말하다'와 '결론을 명시적으로 제시하다'에 해당하는 부사를 찾는 문제이다.

NoteBase

핵심 개념

directly는 중간 과정이나 암시 없이 '직접적으로'라는 뜻이고, indirectly는 '간접적으로'라는 뜻이다. explicitly는 뜻을 숨기지 않고 '명시적으로'라는 뜻이며, implicitly는 말로 분명히 드러내지 않고 '암시적으로'라는 뜻이다. 지문에서 정신과 의사는 환자에게 원인을 스스로 발견하게 하며 직접 말하지 않는다. 반면 청중이 잘못된 결론을 내릴 위험을 피하는 안전한 방법은 결론을 명시적으로 밝히는 것이다.

선지별 해설

① O — (A)의 tell them directly는 환자에게 원인을 '직접 말하다'라는 뜻이고, (B)의 state conclusions explicitly는 결론을 '명시적으로 진술하다'라는 뜻이다. 두 빈칸 모두 문맥에 맞는다.
② X — (B)의 implicitly가 틀렸다. 지문은 결론을 암시하는 전략이 청중의 오해를 낳을 수 있다고 한 뒤 더 안전한 방법을 제시하므로, (B)는 implicitly가 아니라 explicitly가 되어야 한다. 올바른 조합은 directly / explicitly이다.
③ X — (A)의 indirectly가 틀렸다. 환자가 원인을 스스로 발견하게 하는 것은 의사가 원인을 직접 말하는 것과 대비되므로, tell them 뒤에는 indirectly가 아니라 directly가 와야 한다. 올바른 조합은 directly / explicitly이다.
④ X — (A)의 indirectly와 (B)의 implicitly가 모두 틀렸다. (A)는 '스스로 발견하게 하기보다 직접 말하다'라는 대비이므로 directly이고, (B)는 암시의 위험을 피하여 결론을 분명히 밝힌다는 뜻이므로 explicitly이다. 올바른 조합은 directly / explicitly이다.

오답 포인트

앞부분에서 청중이 스스로 결론을 내리게 하는 방식을 긍정적으로 소개했다는 이유만으로 (B)를 implicitly로 고르기 쉽다. 그러나 however 뒤에서는 그 방식의 위험을 지적하고 있으며, The safer approach는 그 위험을 피하는 반대 방식인 explicitly를 요구한다.

다시 나오면 이렇게 판단

rather than에서는 앞뒤의 대조 대상을 확인한다. (A)는 discover for themselves와 대비되는 '직접 말함'을 찾는다. however와 safer approach에서는 논지가 전환되므로, (B)는 앞서 비판한 imply와 반대되는 '명시적으로 진술함'을 찾는다.

암기 포인트

directly ↔ indirectly는 전달 방식의 대비이고, explicitly ↔ implicitly는 표현의 명시성 여부에 관한 대비이다.

NoteBase