

2018-서울시-컴퓨터일반-A형-해설

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

해설에 대한 모든 권리는 곽후근(대방고시, 하이클래스)에 있습니다.

1. 중위 표기법으로 표현된 <보기>의 수식을 후위 표기법으로 옳게 표현한 것은?

<보기> $a + (b * c - d) * (e - f * g) - h$

① $ab * cd + efg * - * - h -$

② $abc * d + ef * g - * - h -$

③ $abcd * - efg * + * - h -$

④ $abc * d - efg * - * + h -$

정답 체크 :

(4)

중위표기식을 후위표기식으로 바꾸는 방법은 다음과 같다.

피연산자를 만나면 그대로 출력한다.
연산자를 만나면 스택에 저장했다가 스택보다 우선 순위가 같거나 낮은 연산자가 나오면 그때 출력한다.
왼쪽 괄호는 우선순위가 가장 낮은 연산자로 취급한다.
오른쪽 괄호가 나오면 스택에서 왼쪽 괄호위에 쌓여있는 모든 연산자를 출력한다.

해당 방법을 기반으로 중위를 후위로 바꾸면 다음과 같다.

a를 출력한다. 출력 : a

+를 스택에 push한다. 출력 : a

(를 스택에 넣는다. 출력 : a

b를 출력한다. 출력 : ab

*를 스택에 push한다. 출력 : ab

c를 출력한다. 출력 : abc

-와 스택의 최상위인 *를 비교한다. *의 우선순위가 높으므로 출력하고 -는 push한다. 출력 : abc* → (2) 혹은 (4)번이 답임을 알 수 있다.

d를 출력한다. 출력 : abc*d

)는 (위의 모든 연산자를 출력한다. 출력 : abc*d- → 해당 결과로 (4)번이 답임을 알았으므로 더 이상의 진행은 무의미하다.

2. 소프트웨어 개발 프로세스 모델에 대한 설명으로 가장 옳지 않은 것은?

① 폭포수(Waterfall) 모델은 단계별 정형화된 접근 방법 및 체계적인 문서화가 용이하다.

② RAD(Rapid Application Development) 모델은 CASE (Computer Aided Software Engineering) 도구를 활용 하여 빠른 개발을 지향한다.

③ 나선형(Spiral) 모델은 폭포수(Waterfall) 모델과 원형 (Prototype) 모델의 장점을 결합한 모델이다.

④ 원형(Prototype) 모델은 고객의 요구를 완전히 이해하여 개발을 진행하는 것으로 시스템 이해도가 높은 관리자가 있는 경우 유용하다.

정답 체크 :

(4) 원형(Prototype) : 원형을 이용해서 고객의 요구를 더 정확히 추출하는 모델로 고객의 요구를 완전히 이해하여 개발을 진행하는 것이 아니다. 만약, 고객의 요구를 완전히 이했다면 원형 보다는 폭포수를 사용하는 것이 좋다.

오답 체크 :

(1) 폭포수(Waterfall) : 중간 산출물이 명확하고 관리하기가 좋다. 체계적인 문서화가 가능하지만 소용없는 다종의 문서를 생산할 가능성이 있다. 이미 잘 알고 있는 문제나 연구 중심 문제에 적합하다.

(2) RAD(Rapid Application Development) : 프로토타이핑(원형) 모델 기준 사용자 요구사항, 분석, 설계, 개발을 신속한 시스템으로 개발한다(원형 모델을 개선). 제한된 범위의 단독 시스템을 CASE (Computer Aided Software Engineering, 시스템 개발 방법론들의 자동화를 지원하는 소프트웨어 도구를 제공해 개발자의 반복적인 작업량을 줄이도록 하는 것)와 같은 다양한 도구를 활용하여 신속히 개발한다.

(3) 나선형(Spiral) : 폭포수 모델의 장점(중간 산출물이 명확)과 원형 모델의 장점(고객의 요구 반영)을 결합한 모델이다.

3. 서로 다른 시스템 간의 통신을 위한 표준을 제공함으로써 통신에 방해가 되는 기술적인 문제점을 제거하고 상호 인터 페이스를 정의한 OSI 참조 모델의 계층에 대한 설명으로 가장 옳지 않은 것은?

① 네트워크 계층은 물리 계층에서 전달받은 데이터에 대한 동기를 확인하는 기능, 데이터의 원활한 전송을 제어하는 흐름제어(Flow Control) 기능, 안전한 데이터 전송을 위한 에러 제어(Error Control) 기능을 수행한다.

② 물리 계층은 상위 계층으로부터 전달받은 데이터의 물리 적인 링크를 설정하고 유지, 해제하는 기능을 담당한다.

③ 전송 계층은 통신하고 있는 두 사용자 사이에서 데이터 전송의 종단 간(end-to-end) 서비스 질을 높이고 신뢰성을 제어하는 기능을 담당한다.

④ 응용 계층은 사용자가 직접 접하는 부분이며 전자 메일 서비스, 파일 전송 서비스, 네트워크 관리 등이 있다.

정답 체크 :

(1) 네트워크 계층 : 해당 설명은 데이터 링크층에 대한 설명이고, 네트워크 계층은 패킷 종단(End-to-End) 전달, 논리적인(IP) 주소 지정, 경로 지정(Routing), 주소 변환(ARP) 등을 수행한다.

오답 체크 :

(2) 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) 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.

Tip! : OSI 7 계층을 테이블로 정리하면 다음과 같다.

물리 계층(1)	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.
데이터 링크 계층(2)	패킷 노드(Node-to-Node or Peer-to-Peer) 전달, 물리적인(MAC) 주소 지정, 접근 제어(MAC filtering), 흐름 제어(stop-and-wait, sliding window), 오류 처리(ARQ) 등을 수행한다.
네트워크 계층(3)	패킷 종단(End-to-End) 전달, 논리적인(IP) 주소 지정, 경로 지정(Routing), 주소 변환(ARP) 등을 수행한다.
전송 계층(4)	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.
세션 계층(5)	세션 관리, 동기화, 원활한 종료(모든 전송을 마치고 종료) 등을 수행한다.
표현 계층(6)	변환, 암호화/복호화, 압축, 보안을 수행한다.
응용 계층(7)	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. <보기> C 프로그램의 실행 결과는?

```
<보기>
#include <stdio.h>
int main( )
{
    int a=0, b=1;
    switch(a) {
        case 0 : printf( " %d \ n " , b++); ; break;
        case 1 : printf( " %d \ n " , ++b); ; break;
        default : printf( " 0 \ n " , b); ; break;
    }
    return 0;
}
```

- ① 0
- ② 1
- ③ 2
- ④ 3

정답 체크 :

(2)

switch(a) // a가 0이기 때문에 case 0이 선택된다.

printf("%d\n", b++); ; break; // b++는 b의 값 1을 출력하고 나중에 2로 증가한다.

5. 정책 수립에 있어 중요성이 커지고 있는 빅데이터에 대한 설명으로 가장 옳지 않은 것은?

- ① 디지털 환경에서 생성되는 데이터로 규모가 방대하고, 생성 주기가 길며, 형태가 다양하다.
- ② 하둡(Hadoop)과 같은 오픈 소스 소프트웨어 시스템을 빅데이터 처리에 이용하는 것이 가능하다.
- ③ 보건, 금융과 같은 분야의 빅데이터는 사회적으로 유용한 정보이나 데이터 활용 측면에서 프라이버시 침해에 대한 대비가 필요하다.
- ④ 구글 및 페이스북, 아마존의 경우 이용자의 성향과 검색 패턴, 구매패턴을 분석해 맞춤형 광고를 제공하는 등 빅데이터의 활용을 증대시키고 있다.

해설)

정답 체크 :

(1) 아래의 그림은 빅데이터의 속성이다. 데이터 규모가 방대하고 형태가 다양하지만, 생성 주기가 길지 않다.

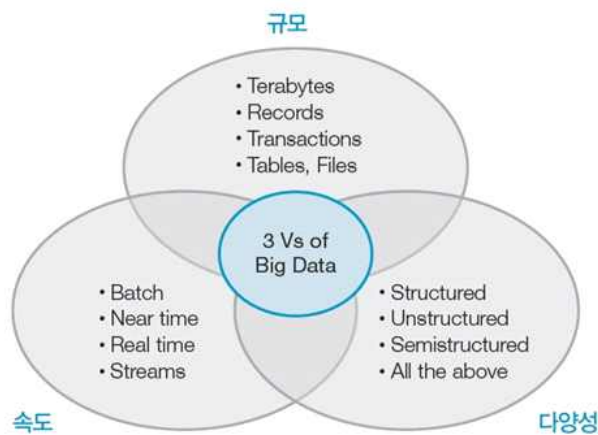


그림 11-24 빅데이터의 속성 (출처: TDWI Research, 2011)

오답 체크 :

- (2) 하둡 : 대량의 자료를 처리할 수 있는 큰 컴퓨터 클러스터에서 동작하는 분산 응용 프로그램을 지원하는 프리웨어 자바 소프트웨어 프레임워크이다.
- (3) 보건, 금융 : 보건, 금융에는 개인 정보가 있으므로 이를 빅데이터로 활용할 때에는 프라이버시 침해에 대한 고려를 해야한다.
- (4) 구글, 페이스북, 아마존 : 아마존의 경우 모든 고객들의 구매내역을 데이터베이스에 기록하고, 이 기록을 분석해 소비자의 소비취향과 관심사를 파악한다. 이런 빅데이터의 활용을 통해 아마존은 고객별로 '추천상품'을 표시한다.

6. <보기>는 TCP/IP 프로토콜에 대한 설명이다. ㉠~㉢에 들어갈 내용으로 가장 옳은 것은?

<보기>	
• (㉠)는 사용자가 입력한 IP 주소를 이용해 물리적 네트워크 주소(MAC Address)를 제공한다.	
• (㉢)는 데이터 전송 과정에서 오류가 발생하면 오류 메시지를 전송한다.	

- | | |
|-------------|---|
| ㉠ | ㉢ |
| ① ICMP RARP | |
| ② RARP ICMP | |
| ③ ARP ICMP | |

④ ICMP ARP

정답 체크 :

(3)

(ㄱ) ARP : IP 주소(논리 주소)를 MAC 주소(물리 주소)로 바꿔준다.

(ㄴ) 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.

오답 체크 :

(1), (2) RARP : MAC 주소(물리 주소)를 IP 주소(논리 주소)로 바꿔준다.

(4) ICMP, ARP의 순서가 바뀌었다.

7. 주기억 장치의 페이지 교체 기법에 대한 설명으로 가장 옳은 것은?

① FIFO(First In First Out)는 가장 오래된 페이지를 교체한다.

② MRU(Most Recently Used)는 최근에 적게 사용된 페이지를 교체한다.

③ LRU(Least Recently Used)는 가장 최근에 사용한 페이지를 교체한다.

④ LFU(Least Frequently Used)는 최근에 사용빈도가 가장 많은 페이지를 교체한다.

정답 체크 :

(1) FIFO : 가장 먼저 들어온 페이지를 교체한다. 즉, 가장 오래된 페이지를 교체한다.

오답 체크 :

(2) MRU : 가장 최근에 사용한 페이지를 교체한다.

(3) LRU : 가장 최근에 사용하지 않은 페이지를 교체한다.

(4) LFU : 가장 사용 빈도가 적은 페이지를 교체한다.

8. RAID(Redundant Array of Inexpensive Disks) 기술에 대한 설명으로 가장 옳지 않은 것은?

① RAID 1 레벨은 미러링(Mirroring)을 지원한다.

② RAID 3 레벨은 데이터를 블록 단위로 분산 저장하여 대용량의 읽기 중심 서버용으로 사용한다.

③ RAID 5 레벨은 고정적인 패리티 디스크 대신 패리티가 모든 디스크에 분산되어 저장되므로 병목 현상을 줄여준다.

④ RAID 6 레벨은 두 개의 패리티 디스크를 사용하므로 두 개의 디스크 장애 시에도 데이터의 복구가 가능하다.

정답 체크 :

(2) RAID 3 : 해당 설명은 RAID 4이고, RAID 3은 바이트 단위이다. RAID 3은 추가 볼륨의 단점(RAID 2)을 조금 더 개선하여, 오직 1개의 볼륨만으로 패리티 정보를 저장할 수 있어 볼륨의 추가 비용이 적게 든다.

오답 체크 :

(1) RAID 1 : 여분의 디스크가 포함되지 않지만 동일한 RAID 볼륨을 추가적으로 구성된다. 추가된 볼륨이 원래의 볼륨과 동일하기 때문에 미러링(mirroring) 모드라고 한다.

(3) RAID 5 : 패리티 비트를 저장하는 볼륨을 별도로 설치(레벨 4)하지 않고, 데이터를 저장

하는 볼륨에 패리티 비트를 분산하여 저장한다.

(4) RAID 6 : 신뢰성에 좀 더 기반을 둔 구성이다. RAID 5에서는 2개의 볼륨에서 동시에 오류가 발생할 경우 복구하기 힘들지만 RAID 6은 패리티 정보를 하나 더 추가해서 동시에 오류가 발생해도 복구가 가능하다.

9. 질의 최적화를 위한 질의문의 내부 형태 변화에 대한 규칙으로 가장 옳지 않은 것은?

① 실렉트(select) 연산은 교환적이다:

$$\sigma_{c1}(\sigma_{c2}(R)) \equiv \sigma_{c2}(\sigma_{c1}(R))$$

② 연속적인 프로젝트(project) 연산은 첫 번째 것을 실행 하면 된다:

$$\Pi_{List1}(\Pi_{List2}(\dots(\Pi_{Listn}(R))\dots)) \equiv \Pi_{Listn}(R)$$

③ 합집합(U)과 관련된 프로젝트(project) 연산은 다음과 같이 변환된다:

$$\Pi(A \cup B) \equiv \Pi(A) \cup \Pi(B)$$

④ 실렉트의 조건 c가 프로젝트 속성만 포함하고 있다면 교환적 이다:

$$\sigma_c(\Pi(R)) \equiv \Pi(\sigma_c(R))$$

정답 체크 :

(2) 연속적인 프로젝트 연산은 마지막 것을 실행한다. Tip!에 정리된 R3에 해당된다.

오답 체크 :

(1) 실렉트 연산은 교환적이다. Tip!에 정리된 R2 규칙에 해당된다.

(3) 합집합과 관련된 프로젝트 연산은 다음과 같이 변환된다. Tip!에 정리된 R11 규칙에 해당된다.

(4) 실렉트의 조건 c가 프로젝트 속성만 포함하고 있다면 교환적이다. Tip!에 정리된 R4에 해당된다.

Tip! : 순수 관계 연산자, 일반 집합 연산자, 그리고 질의 최적화를 위한 질의문의 내부 형태 변화에 대한 규칙을 테이블로 정리하면 다음과 같다.

표 6-2 순수 관계 연산자의 종류

연산자	기호	표현	의미
선택	σ	$\sigma_{조건}(R)$	릴레이션 R에서 조건을 만족하는 튜플들을 반환
프로젝트	π	$\pi_{속성리스트}(R)$	릴레이션 R에서 주어진 속성들의 값으로만 구성된 튜플들을 반환
조인	\bowtie	$R \bowtie S$	공통 속성을 이용해 릴레이션 R과 S의 튜플들을 연결하여 만들어진 새로운 튜플들을 반환
다비전	\div	$R \div S$	릴레이션 S의 모든 튜플과 관련이 있는 릴레이션 R의 튜플들을 반환

표 6-1 일반 집합 연산자의 종류

연산자	기호	표현	의미
합집합	U	$R \cup S$	릴레이션 R과 S의 합집합을 반환
교집합	\cap	$R \cap S$	릴레이션 R과 S의 교집합을 반환
차집합	-	$R - S$	릴레이션 R과 S의 차집합을 반환
카티션 프로덕트	\times	$R \times S$	릴레이션 R의 각 튜플과 릴레이션 S의 각 튜플을 모두 연결하여 만들어진 새로운 튜플을 반환

R1. 논리곱으로 연결된 선택조건 → 일련의 개별적인 선택 조건

$$s_{c_1 \text{ AND } c_2 \text{ AND } \dots \text{ AND } c_n}(R) \equiv s_{c_1}(s_{c_2}(\dots(s_{c_n}(R))\dots))$$

R2. 선택연산은 교환적

$$s_{c_1}(s_{c_2}(R)) \equiv s_{c_2}(s_{c_1}(R))$$

R3. 연속적인 프로젝트 연산(P) → 마지막 것만 실행

$$P_1(P_2(\dots(P_n(R))\dots)) \equiv P_1(R)$$

R4. 셀렉트의 조건 c가 프로젝트 애트리뷰트만 포함하고 있다면 이들은 교환적

$$s_c(P(R)) \equiv (P(s_c(R)))$$

R5. 셀렉트의 조건이 카티션 프로덕트(\times)에 관련된 릴레이션 하나에만 국한 → 조인조건

$$s_c(R \times S) \equiv R \bowtie_c S$$

$$s_{c_1}(R \bowtie_{c_2} S) \equiv R \bowtie_{c_1 \wedge c_2} S$$

R6. 셀렉트의 조건이 조인 또는 카티션 프로덕트에 관련된 릴레이션 하나와만 관련이 되어있을 때

$\sigma_c(R \bowtie S) \equiv \sigma_c(R) \bowtie S$

$\sigma_c(R \times S) \equiv \sigma_c(R) \times S$

R7. c_1 은 릴레이션 R과 관련되어 있고, c_2 는 릴레이션 S와 관련이 되어 있을때 $c = (c_1 \text{ AND } c_2)$

$\sigma_c(R \bowtie S) \equiv (\sigma_{c_1}(R)) \bowtie (\sigma_{c_2}(S))$

$\sigma_c(R \times S) \equiv (\sigma_{c_1}(R)) \times (\sigma_{c_2}(S))$

R8. $\times, \cup, \cap, \bowtie$ 는 교환적

$R \times S \equiv S \times R$

$R \cup S \equiv S \cup R$

$R \cap S \equiv S \cap R$

$R \bowtie S \equiv S \bowtie R$

R9. L_1 은 릴레이션 R에 관련되어있고, L_2 는 릴레이션 S에 관련되어 있을때 $L=(L_1, L_2)$

□ $\Pi_L(R \bowtie S) \equiv (\Pi_{L_1}(R)) \bowtie (\Pi_{L_2}(S))$

□ $\Pi_L(R \times S) \equiv (\Pi_{L_1}(R)) \times (\Pi_{L_2}(S))$

R10. 집합연산과 관련된 선택의 변환

□ $\sigma_c(R \cup S) \equiv \sigma_c(R) \cup \sigma_c(S)$

□ $\sigma_c(R \cap S) \equiv \sigma_c(R) \cap \sigma_c(S)$

□ $\sigma_c(R - S) \equiv \sigma_c(R) - \sigma_c(S)$

R11. 합집합과 관련된 프로젝트의 변환

□ $\Pi(R \cup S) \equiv (\Pi(R)) \cup (\Pi(S))$

R12. $\cup, \cap, \times, \bowtie$ 는 연합적

□ $(R \bowtie S) \bowtie T \equiv R \bowtie (S \bowtie T)$

□ $(R \cup S) \cup T \equiv R \cup (S \cup T)$

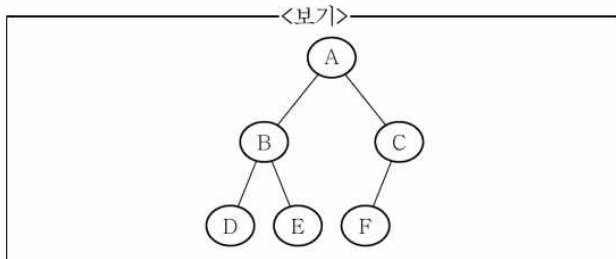
□ $(R \cap S) \cap T \equiv R \cap (S \cap T)$

□ $(R \times S) \times T \equiv R \times (S \times T)$

R13. OR로 연결된 조건식을 AND로 연결된 논리곱 정형식(conjunctive normal form)으로 변환

□ $C_1 \text{ OR } (C_2 \text{ AND } C_3) \rightarrow (C_1 \text{ OR } C_2) \text{ AND } (C_1 \text{ OR } C_3)$

10. <보기> 이진 트리의 내부 경로 길이(length)와 외부 경로 길이로 옳은 것은?



① 7, 20

② 7, 23

③ 8, 20

④ 8, 23

정답 체크 :

(3)

내부 경로 길이 : 루트로부터 각 내부 노드 경로 길이의 합이다. $8 = 1(AB) + 1(AC) + 2(ABD) + 2(ABE) + 2(ACF)$

외부 경로 길이 : 루트로부터 각 외부 노드 경로 길이의 합이다. $20 = 3(ABD\text{왼쪽}) + 3(ABD\text{오른쪽}) + 3(ABE\text{왼쪽}) + 3(ABE\text{오른쪽}) + 3(ACF\text{왼쪽}) + 3(ACF\text{오른쪽}) + 2(AC\text{오른쪽})$

Tip! : (외부 경로 길이 = 내부 경로 길이 + 2 x 노드의 개수)라는 수식이 성립한다. 즉, 외부 경로 길이 = $8 + 2 \times 6 = 20$ 이 된다.

11. 8진수로 표현된 $13754_{(8)}$ 를 10진수로 표현하면?

- ① 6224
- ② 6414
- ③ 6244
- ④ 6124

정답 체크 :

(4)

$$13754_{(8)} = 1 \times 8^4 + 3 \times 8^3 + 7 \times 8^2 + 5 \times 8^1 + 4 \times 8^0 = 6124_{(10)}$$

12. <보기> 잘 알려진 포트번호(well-known port)와 TCP 프로토콜이 바르게 연결된 것을 모두 고른 것은?

<보기>

ㄱ: 21번 포트 : FTP
ㄴ: 53번 포트 : TELNET
ㄷ: 23번 포트 : SMTP
ㄹ: 80번 포트 : HTTP

- ① ㄱ, ㄴ
- ② ㄱ, ㄹ
- ③ ㄴ, ㄷ
- ④ ㄴ, ㄹ

정답 체크 :

(2)

(ㄱ) 21번 포트 : FTP

(ㄹ) 80번 포트 : HTTP

오답 체크 :

(ㄴ) 53번 포트 : DNS, TELNET는 23번 포트이다.

(ㄷ) 23번 포트 : TELNET, SMTP는 25번 포트이다.

Tip! : 주표 포트와 서비스를 테이블로 정리하면 다음과 같다.

포트 번호	서비스	설명
20	FTP	• File Transfer Protocol-Datagram • FTP 연결 시 실제로 데이터를 전송한다.
21	FTP	• File Transfer Protocol-Control • FTP 연결 시 인증과 제어를 한다.
23	Telnet	• 텔넷 서비스로, 원격지 서버의 실행장을 얻어낸다.
25	SMTP	• Simple Message Transfer Protocol • 메일을 보낼 때 사용한다.
53	DNS	• Domain Name Service • 이름을 해석하는 데 사용한다.
69	TFTP	• Trivial File Transfer Protocol • 인증이 존재하지 않는 단순한 파일 전송에 사용한다.
80	HTTP	• Hyper Text Transfer Protocol • 웹 서비스를 제공한다.
110	POP3	• Post Office Protocol • 메일 서버로 전송된 메일을 읽을 때 사용한다.
111	RPC	• Sun의 Remote Procedure Call • 원격에서 서버의 프로세스를 실행할 수 있게 한다.
138	NetBIOS	• Network Basic Input Output Service • 윈도우에서 파일을 공유할 수 있게 한다.
143	IMAP	• Internet Message Access Protocol • POP3와 기본적으로 같으나, 메일이 확인된 후에도 서버에 남는다는 것이 다르다.
161	SNMP	• Simple Network Management Protocol • 네트워크 관리와 모니터링을 위해 사용한다.

13. 파일 처리 시스템(File Process System)과 비교한 데이터 베이스관리 시스템(DBMS)에 대한 설명으로 가장 옳지 않은 것은?

- ① 응용 프로그램과 데이터 간의 상호 의존성이 크다.
- ② 데이터 중복을 최소화한다.
- ③ 응용 프로그램의 요청을 수행한다.
- ④ 데이터 공유를 수월하게 한다.

정답 체크 :

(1) 응용 프로그램과 데이터 간의 상호 의존성이 작다. 데이터를 수정하면 응용 프로그램을 수정할 필요 없이 DBMS에서 처리를 해주면 된다.

오답 체크 :

- (2) 데이터베이스를 이용해서 데이터 중복을 최소화한다.
- (3) 응용 프로그램의 요청을 받아 데이터베이스에게 전달한다. 또한 데이터베이스의 응답(출력 결과)를 사용자에게 전달한다.
- (4) 데이터베이스를 이용해서 데이터 공유를 수월하게 할 수 있다.

Tip! : DBMS에서의 데이터 관리를 그림으로 나타내면 다음과 같다.

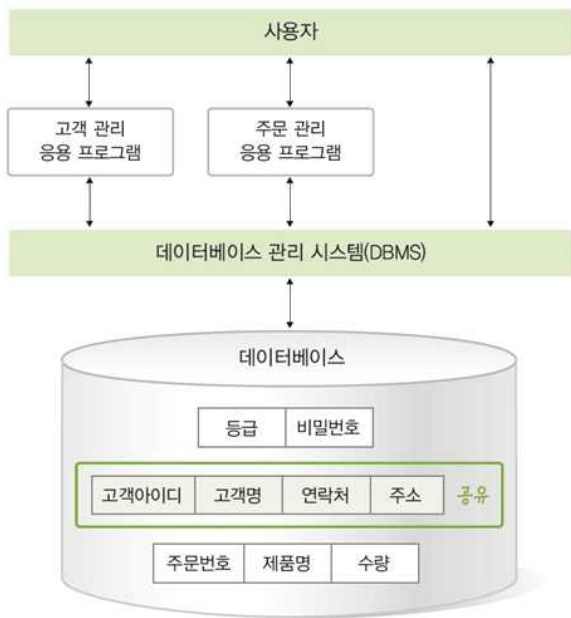


그림 2-5 데이터베이스 관리 시스템에서의 데이터 관리

14. 임계지역(critical section) 문제에 대한 해결책이 가져야 하는 성질로 가장 옳지 않은 것은?

- ① 한 번에 한 프로세스만이 임계지역을 수행하도록 해야 한다.
- ② 프로세스는 자신이 임계지역을 수행하지 않으면서 다른 프로세스가 임계지역을 수행하는 것을 막으면 안된다.
- ③ 프로세스의 임계지역 진입은 유한 시간 내에 이루어져야 한다.
- ④ 임계지역 문제의 해결책에서는 프로세스의 수행 속도에 대해 적절한 가정을 할 수 있다.

정답 체크 :

(4) 프로세스의 속도나 프로세서 수에 영향 받지 않는다.

오답 체크 :

- (1) 두 프로세스는 동시에 공유 자원에 진입 불가이다.
- (2) 공유 자원을 사용하는 프로세스만 다른 프로세스 차단 가능하다.
- (3) 프로세스가 공유 자원을 사용하려고 너무 오래 기다려서는 안된다.

15. <보기> C 프로그램의 출력은?

```

<보기>
#include <stdio.h>
int main()
{
    int a = 5, b = 5;
    a *= 3 + b++;
    printf("%d %d" , a, b) ;
    return 0;

```

```
}
```

- ① 40 5
- ② 40 6
- ③ 45 5
- ④ 45 6

정답 체크 :

(2)

int a = 5, b = 5;

a = a * (3 + b++); // a = 5 * (3 + 5) = 40;

b = 6; // b++는 후위 연산으로 위의 연산이 끝난 후에 증가한다.

16. <보기> 회로의 종류를 바르게 연결한 것은?

<보기>
ㄱ. 3개의 입력 중에서 적어도 2개의 입력이 1이면 출력이 1이 되는 회로
ㄴ. 설정된 값이 표시되었을 때, 경고음을 울리는 카운터

- | | | |
|---|--------|--------|
| | ⌋ | ⌌ |
| ① | 조합논리회로 | 조합논리회로 |
| ② | 조합논리회로 | 순차논리회로 |
| ③ | 순차논리회로 | 조합논리회로 |
| ④ | 순차논리회로 | 순차논리회로 |

정답 체크 :

(2)

(ㄱ) 조합논리회로 : 현재의 입력이 현재의 출력을 결정한다. 즉, 과거의 출력에 영향을 받지 않는다. 문제의 주어진 조건을 보면 현재의 출력이 과거의 출력에 영향을 받지 않으므로 조합논리회로이다.

(ㄴ) 순차논리회로 : 현재의 입력과 과거의 출력이 현재의 출력을 결정한다. 즉, 과거의 출력에 영향을 받는다. 카운터의 경우 현재값이 10이 되려면 과거값 9와 현재값 1이 있어야 하므로 순차논리회로이다.

17. CISC(Complex Instruction Set Computer)에 대한 설명으로 가장 옳은 것은?

- ① 고정 길이의 명령어 형식을 가진다.
- ② 명령어의 길이가 짧다.
- ③ 다양한 어드레싱 모드를 사용한다.
- ④ 하나의 명령어로 복잡한 명령을 수행할 수 없어 복잡한 하드웨어가 필요하다.

정답 체크 :

(3) CISC는 다양한 어드레싱 모드(주소 기법)를 사용하고, RISC는 간단한 어드레싱 모드를 사용한다. RISC는 레지스터를 많이 사용하므로 레지스터 주소 지정 방식을 많이 사용한다.

오답 체크 :

- (1) CISC는 가변 길이의 명령어 형식을 가지고, RISC는 고정 길이의 명령어 형식을 가진다.
- (2) CISC는 명령어의 길이가 길고 프로그램의 길이가 짧다. 반대로, RISC는 명령어의 길이가 짧고 프로그램의 길이가 길다.

(3) CISC는 하나의 명령어로 복잡한 명령어를 수행할 수 있기 때문에 복잡한 하드웨어(제어 장치)가 필요하다. 반대로, RISC는 하나의 명령어로 복잡한 명령어를 처리할 수 없기 때문에 간단한 하드웨어(제어 장치)를 사용한다. 제어 장치를 만드는 비용을 아껴서 CISC에 비해 많은 레지스터를 사용한다.

18. 퀵 정렬에 대한 설명으로 가장 옳지 않은 것은?

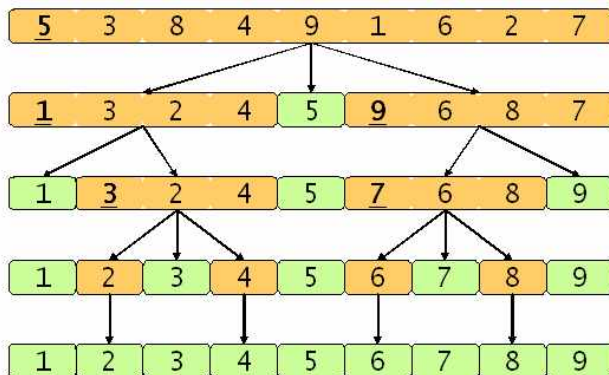
- ① 퀵 정렬은 분할 정복(divide and conquer) 방식으로 동작한다.
- ② 퀵 정렬의 구현은 흔히 재귀 함수 호출을 포함한다.
- ③ n개의 데이터에 대한 퀵 정렬의 평균 수행 시간은 $O(\log n)$ 이다.
- ④ C.A.R. Hoare가 고안한 정렬 방식이다.

정답 체크 :

(3) 평균 수행 시간 : 패스의 수가 $\log n$ 이고, 각 패스 안에서의 비교 횟수는 n 이므로 평균 수행 시간은 $O(n \log n)$ 이다.

오답 체크 :

(1) 분할 정복 : 퀵 정렬은 다음과 같은 분할 정복법을 사용한다. 밑줄친 숫자를 피벗을 나타낸다. 피벗(pivot)을 가장 왼쪽 숫자라고 가정하고, 두 개의 변수 low와 high를 사용한다. low는 피벗보다 작으면 통과하고 크면 정지한다. 그리고 high는 피벗보다 크면 통과하고 작으면 정지한다. 정지된 위치의 숫자를 교환하고 low와 high가 교차하면 종료한다.



(2) 재귀 함수 호출 : 리스트를 2개의 부분리스트로 비균등 분할하고, 각각의 부분리스트를 다시 퀵정렬함으로써 재귀 함수 호출을 사용한다.

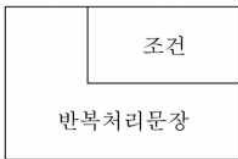
```
void quick_sort(int list[], int left, int right)
{
    if (left < right) {
        int q = partition(list, left, right); // (1)번 지문에서 기술한 알고리즘 사용
        quick_sort(list, left, q-1); // 재귀 함수 호출(피벗 왼쪽 리스트)
        quick_sort(list, q+1, right); // 재귀 함수 호출(피벗 오른쪽 리스트)
    }
}
```

(4) C.A.R Hoare : 퀵 정렬은 토니 호어가 고안하였다. 아래의 그림은 토니 호어를 나타낸다.

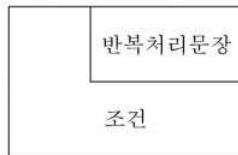


19. 나시-슈나이더만(N-S) 차트의 반복(While) 구조에 대한 표현으로 가장 옳은 것은?

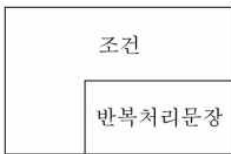
①



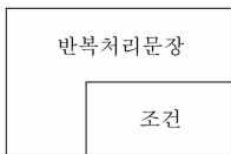
②



③



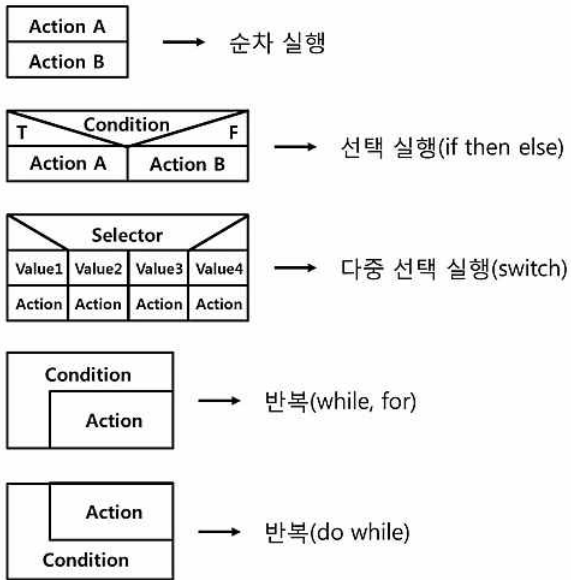
④



정답 체크 :

(3)

나시-슈나이더만 차트는 사각형 박스로 선택, 조건, 반복의 구조적 흐름을 표현한다. 장점은 표준화가 가능하다는 점이고, 단점은 알고리즘이 복잡하면 수직적으로 확장한다는 것이다. 다음은 나시-슈나이더만 차트를 나타낸다. 4번째 그림을 보면 반복이 Condition(조건)과 Action(반복처리문장)으로 구성됨을 알 수 있다.



20. <보기> C 프로그램의 실행 결과로 화면에 출력되는 숫자가 아닌 것은?

```

<보기>
#include <stdio.h>
int my(int i, int j) {
    if (i<3) i=j=1;
    else {
        i=i-1;
        j=j-i;
        printf("%d, %d, ", i, j);
        return my(i,j);
    }
}
int main(void)
{
    my(5,14);
    return 0;
}

```

- ① 1
- ② 3
- ③ 5
- ④ 7

정답 체크 :

my(5, 14); // i = 4, j = 10를 출력하고 my(4, 10)을 호출한다.

my(4, 10); // i = 3, j = 7를 출력하고 my(3, 7)을 호출한다.

my(3, 7); // i = 2, j = 5를 출력하고 my(2, 5)를 호출한다.

my(2, 5); // 2는 3보다 작으므로 순환 호출을 종료한다.

(1) 4번의 순환 호출 과정에서 1을 출력되지 않는다.

오답 체크 :

(2) my(4, 10)에서 출력한다.

(3) my(3, 7)에서 출력한다.

(4) my(4, 10)에서 출력한다.