2016년 계리직 컴퓨터일반 풀이

by 호이호이꿀떡

정답 체크

01	02	03	04	05	06	07	08	09	10
①	③	①	②	④	②	②	③	③	③
11	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
①	④	③	③	①	④	②	①	②	④

문 1. 2진수 11110000과 10101010에 대해 XOR논리 연산을 수행한 결과 값을 16진수로 바르게 표현한 것은?

① 5A ② 6B ③ A5 ④ B6

답 ①

```
    11110000
XOR) 10101010
-------------------------
    01011010
```

이를 16진수로 변환하면, $5A_{(16)}$ 이 된다.

문 2. 무선 네트워크 방식에 대한 설명으로 옳은 것은?

① 블루투스(Bluetooth)는 동일한 유형의 기기 간에만 통신이 가능하다.

② NFC방식이 블루투스 방식보다 최대 전송 속도가 빠르다.

③ NFC방식은 액세스 포인트(access point) 없이 두 장치 간의 통신이 가능하다.

④ 최대 통신 가능거리를 가까운 것에서 먼 순서로 나열하면 Bluetooth < Wi-Fi < NFC < LTE 순이다.

답 ③

③ NFC와 블루투스 모두 액세스 포인트 없이 장치간 통신이 가능하다.

<오답 체크> ① 블루투스는 다른 기기와도 통신이 가능하다. 가장 일반적으로 쓰이는 건 스마트폰과 헤드셋, 스마트폰과 노트북, 노트북과 키보드 등이다.

② 블루투스 방식이 훨씬 빠르다. NFC는 수백Kbps 수준이지만, 블루투스는 이론상 최대 24Mbps, 일반적으로는 최대 2Mbps까지 가능하다.

④ NFC < 블루투스 < Wi-Fi < LTE 순이다.

NFC는 10cm 이내로 거의 접촉하는 정도에서만 사용 가능(대표적으로 모바일 교통카드)

블루투스는 보통 10m 이내(노트북과 헤드셋 또는 스마트폰) 최근엔 40m까지 늘어날 전망이다.

Wi-Fi는 최대 50-100m 이내(강의실이나 지하철 와이파이)

LTE는 수km에서 수 백km까지 가능(기지국과 스마트폰)

문 3. <보기>의 프로세스P1, P2, P3을 시간 할당량(time quantum)이 2인 RR(Round-Robin)알고리즘으로 스케줄링할 때, 평균 응답 시간으로 옳은 것은? (단, 응답시간이란 프로세스의 도착 시간부터 처리가 종료될 때까지의 시간을 말한다. 계산 결과값을 소수점 둘째자리에서 반올림한다)

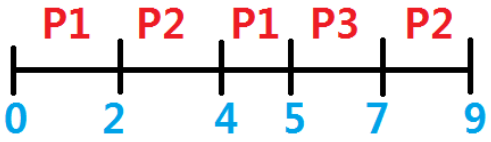
〈보기〉

프로세스	도착시간	실행시간
P1	0	3
P2	1	4
P3	3	2

① 5.7 ② 6.0 ③ 7.0 ④ 7.3

답 ①

▷ RR(Round Robin, 라운드 로빈)
각 프로세스는 할당된 시간(Time Slice Quantum)동안만 실행한 후 실행이 완료되지 않으면 다음 프로세스에게 CPU를 넘겨주고 준비상태 큐의 가장 뒤로 배치함



1. 먼저 P1을 2초간 처리한다.
   중간에 1초에 P2가 도착해 대기열에 등록된다.
2. 2초에 P1은 실행을 멈추고 대기열에서 기다리고 있는 P2에 CPU를 넘겨준 뒤 대기열로 이동한다.
   P2가 실행하는 도중 3초에 P3가 도착해 대기열의 P1 뒤에서 대기한다.
3. 4초에 P2가 실행을 멈추고 대기열 제일 앞에서 기다리고 있는 P1에 CPU를 넘겨준 뒤, 대기열의 P3 뒤로 이동한다.
4. 5초에 P1가 남은 작업을 모두 끝내고 나간 뒤, 대기열 제일 앞에 있는 P3가 CPU를 사용한다.
5. 7초에 P3가 작업을 모두 끝내고 나간 뒤, 대기열에 있는 P2가 CPU를 사용한다.
6. 9초에 P2까지 작업을 모두 끝내고 나간다.
   걸린 시간 평균은 (5 + 8 + 4) / 3 = 17 / 3 ≒ 5.7

프로세스	도착	실행	완료	시간
P1	0	3	5	5
P2	1	4	9	8
P3	3	2	7	4

문 4. 다음 C프로그램의 실행 결과로 옳은 것은?

```
#include 〈stdio.h〉
int main() {
    int a=120, b=45;
    while ( a != b ) {
        if ( a 〉 b) a = a - b;
        else b = b - a;
    }
    printf("%d", a);
}
```

① 5 ② 15 ③ 20 ④ 25

답 ②

처음 변수 값 a=120, b=45로 시작
a와 b가 다르므로 while문 실행
a가 b보다 크므로, a에 a - b = 120 - 45 = 75 대입

a=75, b=45 이므로 while문 실행
a가 크므로, a에 75 - 45 = 30 대입

a=30, b=45 이므로 while문 실행
a가 크지 않으므로, b에 b - a = 45 - 30 = 15 대입

a=30, b=15 이므로 while문 실행
a가 크므로, a에 a - b = 30 - 15 = 15 대입

a=15, b=15 이므로 while문을 실행하지 않고 빠져나온다.
아래 printf문을 통해 **a값 15를 출력**한다.

문 5. <보기>와 같이 수행되는 정렬 알고리즘으로 옳은 것은?

〈보기〉

단계 0 :	6	5	8	9	4	2
단계 1 :	6	5	8	2	4	9
단계 2 :	6	5	4	2	8	9
단계 3 :	2	5	4	6	8	9
단계 4 :	2	4	5	6	8	9
단계 5 :	2	4	5	6	8	9

① 쉘 정렬(shell sort)
② 히프 정렬(heap sort)
③ 버블 정렬(bubble sort)
④ 선택 정렬(selection sort)

답 ④

④ 각 단계를 보면, 아직 정렬되지 않은 값들 중에서 가장 큰 값을 찾아 제일 뒤의 값과 자리를 바꾸면서, 뒤에서부터 큰 수를 하나씩 찾아 채워지고 있다.
이는 선택 정렬 알고리즘으로 볼 수 있다.
보통 선택 정렬은 앞에서부터 작은 값을 채워나가는 방식으로 정렬하나, 문제에서처럼 뒤에서부터 채워나갈 수도 있다.

단계 0 [6 | 5 | 8 | 9 | 4 | 2]
단계 1 : 자료들 중에서 가장 큰 값인 9를 찾아 맨 뒤 2와 자리를 바꾼다.
[6 | 5 | 8 | 2 | 4 | 9]
단계 2 : 정렬되지 않은 6~4까지 중 가장 큰 값인 8을 맨 뒤 4와 자리를 바꾼다.
[6 | 5 | 4 | 2 | 8 | 9]
단계 3 : 정렬되지 않은 6~2까지 중 가장 큰 값인 6을 맨 뒤의 2와 자리를 바꾼다.
[2 | 5 | 4 | 6 | 8 | 9]
단계 4 : 정렬되지 않은 2~4까지 중 가장 큰 값인 5을 맨 뒤의 4와 자리를 바꾼다.
[2 | 4 | 5 | 6 | 8 | 9]
단계 5 : 정렬되지 않은 2와 4중 큰 값인 4가 뒤에 있으므로 가만 놔둔다.
[2 | 4 | 5 | 6 | 8 | 9]

<오답 체크> ① 쉘 정렬(shell sort)
삽입 정렬을 개선한 정렬 방법으로, 일정한 간격만큼 떨어진 값들을 하나의 부분 그룹으로 묶어 정렬시켜 나가는 방법이다.
② 힙 정렬(heap sort)
힙 정렬은 자료들로 최대(최소) 힙 트리를 구성하여 루트 노드에 있는 제일 큰(작은) 값을 하나씩 빼서 정렬하는 방법이다.
최대(최소) 힙 트리 이진 트리 구조로 부모 노드의 값이 자식 노드의 값보다 크거나 같은(작거나 같은) 트리 구조를 가진다.
③ 버블 정렬(bubble sort)
앞에서부터 차례대로 인접한 두 수를 비교해가면서 큰 수를 뒤로 보내는 정렬 방법이다.

문 6. 직원(사번, 이름, 입사년도, 부서)테이블에 대한 SQL 문 중 문법적으로 옳은 것은?

① SELECT COUNT (부서) FROM 직원 GROUP 부서;
② SELECT * FROM 직원 WHERE 입사년도 IS NULL;
③ SELECT 이름, 입사년도 FROM 직원 WHERE 이름 ='최%';
④ SELECT 이름, 부서 FROM 직원 WHERE 입사년도 =(2014,2015);

답 ②

② 옳은 표현이다. 입사년도가 기본키라는 말도 없으므로 NULL 값을 가질 수 있다. NULL 값을 검색할 땐 '=' 기호를 쓰지 않고, 'IS NULL' 표현을 쓴다.
입사년도 값이 비어있는 튜플을 출력한다.

<오답 체크> ① GROUP 부서(X) -> GROUP BY 부서(○)
-> SELECT COUNT (부서) FROM 직원 GROUP BY 부서;
튜플들을 부서가 같은 것끼리 그룹으로 묶은 뒤, 각 부서별로 튜플 개수를 출력한다.

③ 이름 = '최%'(X) -> 이름 LIKE '최%'(○)
문자열를 찾을 땐 '=' 기호가 아닌 'LIKE' 표현을 쓴다.
'%'는 문자열을 찾고, '_'는 한 글자만 찾는다.
-> SELECT 이름,입사년도 FROM 직원 WHERE 이름 LIKE '최%'
이름이 '최'로 시작하는 튜플의 이름과 입사년도를 출력한다.

④ 입사년도 = (2014, 2015)(X)
-> 입사년도=2014 OR 입사년도=2015(○)
-> 입사년도 IN(2014, 2015) (○)
둘 이상의 값 중 하나에 맞는 것을 찾을 땐 'OR' 조건식으로 따로 쓰거나, IN 구문으로 묶는다.
-> SELECT 이름,부서 FROM 직원
WHERE 입사년도 IN(2014, 2015);
입사년도가 2014년이나 2015년인 튜플을 찾아 출력한다.

문 7. <보기>와 같은 특성을 갖는 하드 디스크의 최대 저장 용량은?

〈보기〉
◦ 실린더 (cylinder) 개수: 32,768 개
◦ 면 (surface) 개수: 4개
◦ 트랙 (track) 당 섹터 (sector) 개수: 256 개
◦ 섹터크기 (sector size): 512 bytes

① 4GB ② 16GB ③ 64GB ④ 1TB

답 ②

트랙 당 섹터 수가 256개이고, 섹터 하나의 크기가 512byte 이므로, 하나의 트랙은 256 × 512 = 128KB 이다.
한 면에 있는 트랙의 수는 실린더의 수와 같다.
그러므로 한 면은
32768 × 128KB = 32K × 128KB = 4096MB = 4GB
총 4면으로 구성되어 있으므로 **전체 용량은 16GB**가 된다.

○ 하드디스크 용량
= 헤드 수 × 실린더 수 × 섹터 수 × 섹터당 용량
(위의 식에서 헤드 수는 디스크 면 수와 같다.)

문 8. <보기>의 직원 테이블에서 키(key)와 관련된 설명으로 옳지 않은 것은?(단, 사번과 주민등록번호는 각 유일한 값을 갖고, 부서번호는 부서 테이블을 참조하는 속성이며, 나이가 같은 동명이인이 존재할 수 있다)

〈보기〉
직원(사번, 이름, 주민등록번호, 주소, 나이, 성별, 부서번호)

① 부서번호는 외래키이다.
② 사번은 기본키가 될 수 있다.
③ (이름, 나이)는 후보키가 될 수 있다.
④ 주민등록번호는 대체키가 될 수 있다.

답 ③

③ 후보키는 유일성과 최소성을 만족해야 한다. 나이가 같은 동명이인이 존재할 수 있으므로, (이름,나이) 속성은 유일성을 만족하지 못해 후보키가 될 수 없다.

<오답 체크> ① 부서번호는 다른 테이블 속성을 참조하므로 외래키이다.
② 사번은 유일성과 최소성을 만족하므로 기본키로 사용할 수 있다.
④ 주민등록번호는 유일성과 최소성을 만족하므로 기본키로 대용이 가능한 수 있는 대체키가 될 수 있다.

문 9. 소프트웨어 테스트에 대한 설명으로 옳지 않은 것은?

① 베타(beta) 테스트는 고객 사이트에서 사용자에 의해서 수행된다.
② 회귀(regression) 테스트는 한 모듈의 수정이 다른 부분에 미치는 영향을 검사한다.
③ 화이트 박스(white box) 테스트는 모듈의 내부 구현보다는 입력과 출력에 의해 기능을 검사한다.
④ 스트레스(stress) 테스트는 비정상적으로 과도한 분량 또는 빈도로 자원을 요청할 때의 영향을 감사한다.

답 ③

③ 입력값과 출력값을 통해 테스트하는 방식은 블랙박스(black box) 테스트이다.
화이트박스(white box) 테스트는 소프트웨어 내부 구현과 소스코드를 분석하는 테스트 기법이다.

<오답 체크> ① 기업 내부에서 관련자들끼리 자체 테스트하는 방식을 알파(alpha) 테스트, 고객이나 유저 등 외부 사용자들을 대상으로 테스트하는 방식을 베타(beta) 테스트라고 한다.

② **회귀**(regression) 테스트
이미 테스트된 프로그램의 테스트를 반복하는 것으로, 결함 수정 이후 새롭게 발견되거나 이전에 발견되지 않았던 또 다른 결함을 발견하는 테스트
한 모듈의 수정이 다른 모듈이나 전체 시스템에 영향을 미쳐 버그를 발생시키지는 않는지, 기존 버그가 재발하지는 않는지, 기존 버그 수정으로 인해 이전에 없던 새로운 버그가 발생하지는 않는지 등을 테스트한다.

④ **스트레스**(stress) 테스트
시스템이 한계점에 도달할 정도의 과부하 상태에 이를 때 어떻게 작동하는지를 테스트하는 것
시스템의 실패를 확인하고 모니터링하는 과정이 정상적으로 이루어지는지, 민감한 정보나 보안상의 문제가 노출되지 않는지, 장애 조치와 복구 절차가 효과적인지 등을 테스트한다.

문 10. <보기>에 선언된 배열 A의 원소 A[8][7]의 주소를 행 우선(row-major) 순서와 열 우선(column-major) 순서로 각각 바르게 계산한 것은?(단, 첫 번째 원소 A[0][0]의 주소는 1,000이고, 하나의 원소는 1byte를 차지한다.)

〈보기〉
char A[20][30];

	행 우선 주소	열 우선 주소
①	1,167	1,148
②	1,167	1,218
③	1,247	1,148
④	1,247	1,218

답 ③

▶ 행 우선(row-major) 주소 방식
2차원 배열의 메모리 주소에 저장할 때, 1행의 값들을 먼저 다 저장하고 그 다음 2행, 3행, 4행의 값들을 저장하는 방식이다.
A[0][0] ~ A[0][29]을 먼저 저장한 뒤,
A[1][0] ~ A[1][29]을 저장하고, A[2][0] ~ A[2][29]을 저장하는 방식이다.

행 우선 방식으로 A[8][7]의 주소를 찾으면, 배열의 열이 30이므로, 각 행마다 원소는 30개씩 있는 셈이다.
먼저 A[0][0] ~ A[0][29]까지 A[0][?] 행의 원소 30개, A[1][?] 행의 원소 30개, A[2][?] 행의 원소 30개 등...
A[7][?]까지 8행의 원소는 240개이다.
그러므로 A[0][0]의 주소가 1000이므로, A[8][0]의 주소는 1240이다.
그리고 A[8][0]에서 7칸 떨어진 **A[8][7]의 주소는 1247**이 된다.

▶ 열 우선(COLUMN-major) 주소 방식
행 우선 방식의 반대로, 1열의 값들을 먼저 다 저장하고 그 다음 2열, 3열, 4열의 값들을 저장하는 방식이다.
A[0][0] ~ A[19][0]을 먼저 저장한 뒤,
A[0][1] ~ A[19][1]을 저장하고, A[0][2] ~ A[19][2]을 저장하는 방식이다.

열 우선 방식으로 A[8][7]의 주소를 찾으면, 배열의 행이 20이므로, 각 열마다 원소가 20개씩 있는 셈이다.
먼저 A[0][0] ~ A[19][0]까지 A[?][0] 행의 원소 20개, A[?][1] 행의 원소 20개, A[?][2] 행의 원소 20개 등...
A[?][6]까지 7열의 원소는 140이다.
그러므로 A[0][0]의 주소가 1000이므로, A[0][7]의 주소는 1140이다.
그리고 A[0][7]에서 8칸 떨어진 **A[8][7]의 주소는 1148**이 된다.

복잡해 보이지만, 원리를 알면 쉽고 빠르게 계산할 수 있으며, 이것을 공식으로도 계산할 수 있다.

A[m][n]에서 A[0][0]의 주소가 X이고
배열 A[i][j]의 주소를 구한다면,
행 우선 주소 = X + i × n + j
열 우선 주소 = X + j × m + i

행 우선 주소 = 1000 + 8 × 30 + 7 = 1247
열 우선 주소 = 1000 + 7 × 20 + 8 = 1148

문 11. 인터넷 주소 체계인 IPv4와 IPv6의 주소 길이와 주소표시 방법을 각각 바르게 나열한 것은?

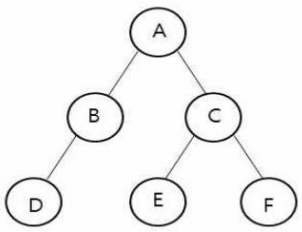
	IPv4	IPv6
①	(32비트, 8비트씩 4부분)	(128비트, 16비트씩 8부분)
②	(32비트, 8비트씩 4부분)	(128비트, 8비트씩 16부분)
③	(64비트, 16비트씩 4부분)	(256비트, 32비트씩 8부분)
④	(64비트, 16비트씩 4부분)	(256비트, 16비트씩 16부분)

답 ①

② IPv4는 8비트씩 4자리, 총 32비트(4바이트)
IPv6는 16비트씩 8자리, 총 128비트(16바이트)

문 12. <보기>의 이진 트리에 대해 지정된 방법으로 순회한 결과가 옳지 않은 것은?

〈보기〉



① 중위순회: D → B → A → E → C → F
② 레벨순회: A → B → C → D → E → F
③ 전위순회: A → B → D → C → E → F
④ 후위순회: D → B → A → E → F → C

답 ④

④ 후위순회(postorder traversal): 왼쪽 자식 노드 → 오른쪽 자식 노드 → 부모 노드 순으로 순회
D → B → E → F → C → A
후위 순회이기 때문에 루트 노드를 제일 나중에 방문한다는 것만 확인해도 바로 틀렸다는 걸 알 수 있다.

<오답 체크> ① 중위순회(inorder traversal): 왼쪽 자식 노드 → 부모 노드 → 오른쪽 자식 노드 순으로 순회

② 레벨순회(level traversal): 모든 노드를 낮은 레벨부터 차례대로 순회하는 것

③ 전위순회(preorder traversal): 부모 노드 → 왼쪽 자식 노드 → 오른쪽 자식 노드 순으로 순회

문 13. 컴퓨터 시스템의 인터럽트(interrupt)에 대한 설명으로 옳지 않은 것은?

① 인터럽트는 입출력 연산, 하드웨어 실패, 프로그램 오류 등에 의해서 발생한다.
② 인터럽트 처리 우선순위 결정 방식에는 폴링(polling) 방식과 데이지 체인(daisy-chain) 방식이 있다.
③ 인터럽트가 추가된 명령어 사이클은 인출 사이클, 인터럽트 사이클, 실행 사이클 순서로 수행된다.
④ 인터럽트가 발생할 경우, 진행 중인 프로그램의 재개(resume)에 필요한 레지스터 문맥(register context)을 저장한다.

답 ③

③ 명령어 처리는 인출(Fetch) - 간접(Indirect) - 실행(Excute) - 인터럽트(Interrupt)의 4단계로 구성된다.
인출 사이클은 기억장치에서 명령어를 중앙처리장치로 읽어오는 사이클, **간접** 사이클은 간접 주소지정 방법을 사용하는 경우에 유효 주소를 찾는 사이클, **실행** 사이클은 명령어를 실행하여 해당 명령어에 맞게 동작을 실행하는 사이클, **인터럽트** 사이클은 인터럽트 발생여부를 파악하고 인터럽트 서브 루틴이 수행하는 사이클이다. 될 수 있도록 해주는 사이클이다.
명령어 사이클 진행중에 인터럽트가 발생해도 실행 사이클까지 모두 끝난 뒤에 인터럽트를 확인하고 처리한다.

<오답 체크> ④ 인터럽트가 발생하면 현재 처리중인 프로세스를 잠시 쉬었다가 인터럽트를 처리하고 다시 현재의 프로세스로 복귀해야 하기 때문에 임시로 필요한 정보들을 저장하는 과정이 필요하다. 이것을 문맥교환이라고 한다.

✠ **컨텍스트 스위칭**(Context Switch, **문맥교환**)
하나의 프로세스가 CPU를 사용 중인 상태에서 다른 프로세스가 CPU를 사용하도록 하기 위해, 이전의 프로세스의 상태(문맥)를 보관하고 새로운 프로세스의 상태를 적재하는 작업을 말한다. 프로세스의 문맥은 프로세스 제어 블록(PCB)에 **기록되어 있다.**

◈ **인터럽트 우선순위 처리 방법**
▷ **소프트웨어적인 판별 방법: 폴링**(Polling)
▷ **하드웨어적인 판별 방법: 벡터 인터럽트**(Vector Interrupt)
- **직렬 방식 : 데이지 체인**(Daisy-chain)
- **병렬 방식**

문 14. 다음은 3년간 연이율 4%로 매월 적립하는 월 복리 정기적금의 만기지급금을 계산한 결과이다. 셀 C2에 들어갈 수식으로 옳은 것은?(단, 만기지급금의 10원 단위 미만은 절사한다)

	A	B	C
1	성명	월적립액	만기지급금
2	김**	₩ 30,000	₩ 1,145,440
3	이**	₩ 50,000	₩ 1,909,070

① =ROUNDDOWN(FV(4%, 3*12, -B2), -1)
② =ROUNDDOWN(FV(4%, 3*12, -B2), -2)
③ =ROUNDDOWN(FV(4%/12, 3*12, -B2), -1)
④ =ROUNDDOWN(FV(4%/12, 3*12, -B2), -2)

답 ③

FV 함수는 투자금에 이자를 더한 원리금 합계를 계산하는 함수로 5개의 인수를 갖고 있는데, 뒤의 인수 2개는 생략이 가능하다.
=FV(기간 이자율, 납입 횟수, 매회 투자금,
(현재 금액), (납입 시기))

연이율은 4%이지만, 매월마다 적립을 하기 때문에 기간 이자율(월이율)은 **4%/12** 이다.
3년간 매월 적립하므로 납입 횟수는 **3*12**회이다.
매월 적립 금액은 30,000원인데, PV 함수에서는 미래의 +금액을 알기 위해서는 - 금액으로 입력해야 한다. 그러므로 **-(B2)**라고 입력한다.
따라서 정기적금의 만기지급금을 계산하는 수식은
=FV(4%/12, 3*12, -B2) 이 된다.

ROUND 함수는 반올림, ROUNDUP 함수는 올림,
ROUNDDOWN 함수는 내림을 하는 함수로, 2개의 인수를 갖는다.
=ROUNDDOWN(값, 표시할 소수점 자릿수)

만기지급금을 10원 단위 미만은 절사한다고 했으므로, 내림 함수인 ROUNDDOWN 함수를 사용한다.
1는 소수점 첫째 자리까지 표시, 0은 정수 일 단위로 표시, -1은 정수 십 단위로 표시한다.
=ROUNDDOWN(123.4567, 1) = 123.4
=ROUNDDOWN(123.4567, 0) = 123
=ROUNDDOWN(123.4567, -1) = 120
10원 단위 미만은 절사한다고 했으므로, 소수점 자릿수로 **-1을 입력**한다.

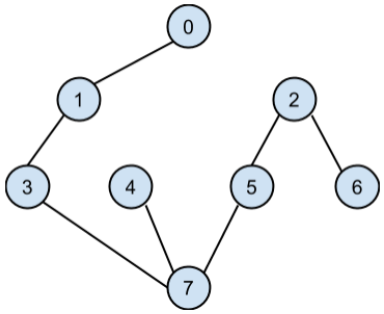
문 15. 정점의 개수가 n 인 연결그래프로부터 생성 가능한 신장트리(spamming tree)의 간선의 개수는?

① $n-1$ ② n ③ $\frac{n(n-1)}{2}$ ④ n^2

답 ①

신장트리는 모든 정점(노드)들이 사이클이 발생하지 않도록 간선으로 연결된 트리를 말한다. 사이클이 발생하지 않아야 하므로, 한 정점에서 다른 정점으로 가는 길은 한 가지만 존재한다.

신장트리에서 정점이 n개면, 간선 n-1개로 연결할 수 있다.



위 신장트리는 정점이 8개이며, 간선 7개로 연결되어 있다.

문 16. <보기>는 관계형 데이터베이스의 정규화 작업을 설명한 것이다. 제1정규형, 제2정규형, 제3정규형, BCNF를 생성하는 정규화 작업을 순서대로 나열한 것은?

〈보기〉
ㄱ. 결정자가 후보키가 아닌 함수 종속성을 제거한다.
ㄴ. 부분 함수 종속성을 제거한다.
ㄷ. 속성을 원자값만 갖도록 분해한다.
ㄹ. 이행적 함수 종속성을 제거한다.

① ㄱ → ㄴ → ㄷ → ㄹ
② ㄱ → ㄷ → ㄹ → ㄴ
③ ㄷ → ㄱ → ㄴ → ㄹ
④ ㄷ → ㄴ → ㄹ → ㄱ

답 ④

ㄷ. 제1정규형
ㄴ. 제2정규형
ㄹ. 제3정규형
ㄱ. 제4정규형

✠ 데이터베이스 정규화 (DB normalization)
관계형 데이터베이스에서 중복을 최소화 하기 위해서 데이터를 분할해가며 구조화하는 작업.

▶ 제1정규형 (1NF)
도메인이 원자값만으로 되어 있어야 함
▶ 제2정규형 (2NF)
부분 함수적 종속 관계를 제거해야 함
▶ 제3정규형 (3NF)
이행 함수적 종속 (x -> y ->z) 제거
▶ BC(Boyce-codd) 정규형 (BCNF)
결정자이면서 후보키가 아닌 것 제거
▶ 제4정규형
다치 종속 제거
▶ 제5정규형
조인 종속성 이용

문 17. 다음 Java 프로그램의 실행 결과로 옳은 것은?

```
class Division  {
    public static void main(String[] args) {
        int a, b, result;
        a = 3;
        b = 0;
        try  {
            result = a / b;
            System.out.print("A");
        }
        catch (ArithmeticException e) {
            System.out.print("B");
        }
        finally {
            System.out.print("C");
        }
        System.out.print("D");
    }
}
```

① ACD ② BCD ③ ABCD ④ BACD

답 ②

◈ try~catch~finally 문은 먼저 try 블록의 코드을 실행한다.
이 때 try 블록에서 오류가 발생하게 되면 try 블록은 수행을 멈추고 catch 블록으로 이동한다.
catch 블록에서는 try 블록에서 발생한 오류에 따라 catch 블록 코드가 실행된다.
finally 블록은 앞에서 오류 발생 및 실행 여부와 관계없이 최종적으로 실행되는 블록이다.
catch 블록과 finally 블록은 생략이 가능하다.

a=3, b=0 이므로, try 블록에서 a / b 연산을 하려고 하면, 나눗셈 연산 오류 ArithmeticException 오류가 발생하고 try 블록을 중단한다.
catch (ArithmeticException e) 블록은 ArithmeticException 오류가 발생했을 때 실행하는 블록이다. 따라서 catch 블록 내의 System.out.print("B"); 코드가 실행되어 **B를 출력**한다.
그리고 오류 여부랑 상관없는 finally 블록이 실행되어, System.out.print("C"); 코드를 통해 **C를 출력**한다.

그 아래 System.out.print("D"); 코드는 try~catch 문과 상관없는 독립된 코드이므로 실행되여, **D를 출력**한다.

문 18. <보기>는 공개키 암호 방식을 전자 서명(digital signature)에 적용하여 A가 B에게 메시지를 전송하는 과정에 대한 설명이다. ㉠, ㉡에 들어갈 내용으로 옳은 것은?

〈보기〉
(1) A와 B는 개인키와 공개키 쌍을 각각 생성한다.
(2) A는 (㉠)를 사용하여 암호화한 메시지를 B에게 전송한다.
(3) B는 (㉡)를 사용하여 수신된 메시지를 해독한다.

	㉠	㉡
①	A의 개인키	A의 공개키
②	A의 개인키	B의 공개키
③	A의 공개키	B의 개인키
④	B의 공개키	B의 개인키

답 ①

✠ 전자서명에서는 메시지에 전자 서명을 해서 보낼 때 개인키로 전자 서명을 하고, 메시지를 받아 전자 서명을 확인할 때 공개키로 서명을 검증한다.
㉠ 메시지에 A가 전자 서명을 하는 것이기 때문에 **A의 개인키**로 메시지를 암호화한다.
㉡ A가 보낸 게 맞는지 검증을 하는 것이기 때문에 **A의 공개키**로 메시지를 해독(복호화)한다.

문 19. 프로그래밍 언어에 대한 설명으로 옳지 <u>않은</u> 것은?

① Objective-C, Java, C#은 객체지향 언어이다.
② Python은 정적 타이핑을 지원하는 컴파일러 방식의 언어이다.
③ ASP, JSP, PHP는 서버 측에서 실행되는 스크립트 언어이다.
④ XML은 전자문서를 표현하는 확장가능한 표준 마크업 언어이다.

답 ②

② **파이썬**(Python)의 특징
- 간결한 코드
- 들여쓰기로 블록 구분
- 인터프리터 언어
- 멀티 패러다임(절차지향, 객체지향, 함수형)
- 동적 타이핑

<오답 체크> ① 객체지향 언어들: C++, C#, Visual Basic, Java, Objective-C, Ada95, SmallTalk, ASP, 델파이, 파이썬, 액션스크립트 등

③

▶ **클라이언트 측 언어**
Active X Controls
Java Applet
Javascript
DHTML

▶ **서버 측 언어**
CGI
Java Servlet
ASP (Active Server Page)
JSP (Java Server Page)
PHP (Personal Home Page Tools)

④ **XML**(eXtensible Markup Language)
W3C에서 개발한, 여러 특수 목적의 마크업 언어를 만들기 위해 사용하도록 권장되는 다목적 마크업 언어이다. XML은 주로 다른 종류의 시스템, 특히 인터넷에 연결된 시스템끼리 데이터를 쉽게 주고 받을 수 있게 하여 HTML의 한계를 극복할 목적으로 만들어졌다.

문 20. <보기>의 설명에 해당하는 기술로 가장 적절한 것은?

〈보기〉
◦ 서비스 모델은 IaaS, PaaS, SaaS로 구분한다.
◦ 필요한 만큼 자원을 임대하여 사용할 수 있다.
◦ 가상화 기술, 서비스 프로비저닝(Provisioning) 기술, 과금 체계 등을 필요로 한다.

① 빅데이터(bigdata)
② 딥 러닝(deep learning)
③ 사물 인터넷(internet of things)
④ 클라우드 컴퓨팅(cloud computing)

답 ④

④ **클라우드 컴퓨팅**(cloud computing)은 인터넷 기반 컴퓨팅의 일종으로 정보를 자신의 컴퓨터가 아닌 인터넷에 연결된 다른 컴퓨터로 처리하는 기술을 의미한다.

▷ Infrastructure as a service(**Iaas**)
클라우드를 이용해, 서버 자원, IP, 네트워크, 저장 공간, 전력 등 컴퓨팅 인프라를 사용하는 데 필요한 자원을 서비스하는 방식이다. 인터넷 상에서 단순 데이터 저장 기능만 제공하는 클라우드 저장 서버가 여기에 해당한다.
▷ Platform as a service(**Paas**)
서비스를 개발 할 수 있는 안정적인 플랫폼(platform)과 그 환경을 이용하는 응용 프로그램을 개발 할 수 있는 API를 제공하는 서비스 방식이다.
▷ Software as a service(**Saas**)
모든 서비스가 클라우드에서 이뤄지는 서비스 방식으로, 별도의 소프트웨어 설치 없이, 클라우드 환경에서 소프트웨어를 이용할 수 있는 방식이다.
이메일 서비스, MS 오피스 365, 구글 문서 등
▷ Mobile Backend as a Service(**MBaaS**)
모바일 앱개발에 필요한 위치기반서비스, 푸시알림, 포토 콜렉션, 사용자 인증, 소셜(social) 네트워크와의 통합 등, 서버와 통신하는 백엔드를 서비스 형태로 개발자에게 제공해 서버 기술을 몰라도 그 환경에 연결되는 모바일 앱을 만들 수 있도록 지원해주는 서비스 방식

<오답 체크> ① **빅데이터**(bigdata)
기존 데이터베이스 관리도구의 능력을 넘어서는 대량의 정형 또는 비정형의 데이터 집합으로부터 가치를 추출하고 결과를 분석하는 기술

▶ 빅데이터 특징 **3V**
- Volume(규모)
- Velocity(속도)
- Variety(다양성)

여기에 Veracity(정확성)와 Value(가치)를 추가해 **5V**라 부른다.
- Veracity(정확성)
- Value(가치)

② **딥 러닝**(deep learning)
컴퓨터를 인간의 두뇌와 비슷한 모양의 대형 신경망으로 구현하여 기계를 학습시키는 머신 러닝의 하나이다. 대규모 인공 신경망에 학습 알고리즘과 데이터를 공급함으로써, 컴퓨터는 그 데이터를 분석하여 스스로 학습하고 훈련해나간다.

③ **사물 인터넷**(internet of things)
정보 통신 기술을 기반으로 실세계와 가상 세계의 다양한 사물들을 연결하여 진보된 서비스를 제공하기 위한 지능형 인프라 및 서비스 기술이다.