

시험에  
나오는 것만  
공부한다!



## 값만 다르게 매번 출제되는 필수 계산 41문제 정보처리기사 필기



1. 다음 릴레이선의 카디널리티와 차수가 옳게 나타낸 것은?

아이디	성명	나이	등급	적립금	가입년도
yuyu01	원유철	36	3	2000	2008
skkim10	김성일	29	2	3300	2014
kshan4	한경선	45	3	2800	2009
namsu52	이남수	33	5	1000	2016

- ① 카디널리티 : 4, 차수 : 4
- ② 카디널리티 : 4, 차수 : 6
- ③ 카디널리티 : 6, 차수 : 4
- ④ 카디널리티 : 6, 차수 : 6

### [해설]

- 테이블에 속한 튜플의 수를 카디널리티(Cardinality), 속성의 수를 차수(Degree)라고 합니다.
- 카디널리티는 4, 차수는 6입니다.

2. A1, A2, A3 3개 속성을 갖는 한 릴레이션에서 A1의 도메인은 3개 값, A2의 도메인은 2개 값, A3의 도메인은 4개 값을 갖는다. 이 릴레이션에 존재할 수 있는 가능한 튜플(Tuple)의 최대 수는?

- ① 24
- ② 12
- ③ 8
- ④ 9

### [해설]

튜플의 최대 수는 각 도메인의 값을 모두 곱한 값으로  $3 \times 2 \times 4 = 24$ 입니다.

3. 기억공간이 15K, 23K, 22K, 21K 순으로 빈 공간이 있을 때 기억장치 배치 전략으로 “First Fit”을 사용하여 17K의 프로그램을 적재할 경우 내부 단편화의 크기는 얼마인가?

- ① 5K
- ② 6K
- ③ 7K
- ④ 8K

### [해설]

- 최초 적합(First Fit)은 프로그램이나 데이터가 들어갈 수 있는 크기의 빈 영역 중에서 첫 번째 분할 영역에 배치시키는 방법으로, 17K보다 큰 영역 중 첫 번째 영역인 23K에 배치됩니다.
- 내부 단편화는 분할된 영역이 할당될 프로그램의 크기보다 크기 때문에 프로그램이 할당된 후 사용되지 않고 남아 있는 빈 공간을 의미하므로,  $23K - 17K = 6K$ 가 됩니다.

4. 버블 정렬을 이용하여 다음 자료를 오름차순으로 정렬할 경우 PASS 1의 결과는?

9, 6, 7, 3, 5

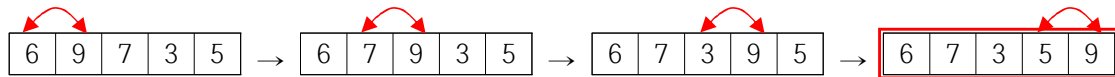
- ① 6, 9, 7, 3, 5
- ② 3, 9, 6, 7, 5
- ③ 3, 6, 7, 9, 5
- ④ 6, 7, 3, 5, 9

[해설]

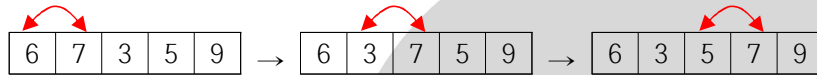
버블 정렬은 주어진 파일에서 인접한 두 개의 레코드 키 값을 비교하여 그 크기에 따라 레코드 위치를 서로 교환하는 정렬 방식으로 다음과 같은 과정으로 진행됩니다.

• 원본 : 9, 6, 7, 3, 5

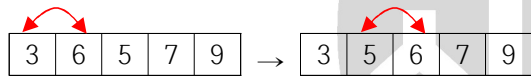
❶ 1회전



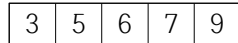
❷ 2회전



❸ 3회전



❹ 4회전



5. 다음 자료에 대하여 선택(Selection) 정렬을 이용하여 오름차순으로 정렬하고자 한다. 1회전 수행 결과는?

8, 3, 4, 9, 7

① 3, 4, 7, 8, 9

② 3, 4, 7, 9, 8

③ 3, 4, 8, 9, 7

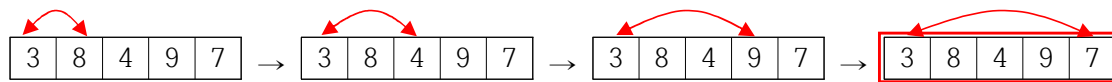
④ 3, 8, 4, 9, 7

[해설]

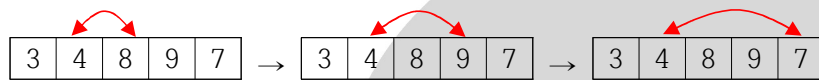
선택 정렬은  $n$ 개의 레코드 중에서 최소값을 찾아 첫 번째 레코드 위치에 놓고, 나머지  $n-1$ 개 중에서 다시 최소값을 찾아 두 번째 레코드 위치에 놓는 방식을 반복하여 정렬하는 방식입니다.

• 원본 : 8, 3, 4, 9, 7

① 1회전



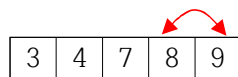
② 2회전



③ 3회전



④ 4회전



6. 다음 초기 자료에 대하여 삽입 정렬(Insertion Sort)을 이용하여 오름차순 정렬할 경우 1회전 후의 결과는?

초기 자료 : 8, 3, 4, 9, 7

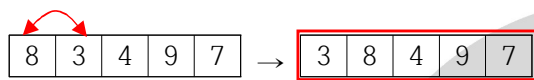
- ① 3, 4, 8, 7, 9
- ② 3, 4, 9, 7, 8
- ③ 7, 8, 3, 4, 9
- ④ 3, 8, 4, 9, 7

**[해설]**

삽입 정렬은 두 번째 자료부터 시작하여 그 앞(왼쪽)의 자료들과 비교하여 삽입할 위치를 지정한 후 자료를 뒤로 옮기고 지정한 자리에 자료를 삽입하여 정렬하는 알고리즘입니다. 즉 두 번째 자료는 첫 번째 자료, 세 번째 자료는 두 번째와 첫 번째 자료, 네 번째 자료는 세 번째, 두 번째, 첫 번째 자료와 비교한 후 자료가 삽입될 위치를 찾습니다.

• 초기 자료 : 8, 3, 4, 9, 7

**① 1회전**



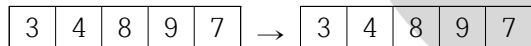
두 번째 값 3을 첫 번째 값과 비교하여 첫 번째 자리에 삽입하고 8을 한 칸 뒤로 이동시킵니다.

**② 2회전**



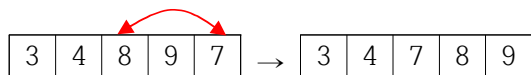
세 번째 값 4를 첫 번째, 두 번째 값과 비교하여 8자리에 삽입하고 8을 한 칸 뒤로 이동시킵니다.

**③ 3회전**



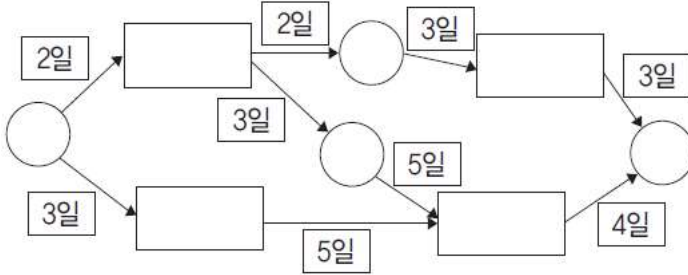
네 번째 값 9를 첫 번째, 두 번째, 세 번째 값과 비교한 후 삽입할 곳이 없다면 다음 회전으로 넘어갑니다.

**④ 4회전**



다섯 번째 값 7을 처음부터 비교하여 8자리에 삽입하고 나머지를 한 칸씩 뒤로 이동시킵니다.

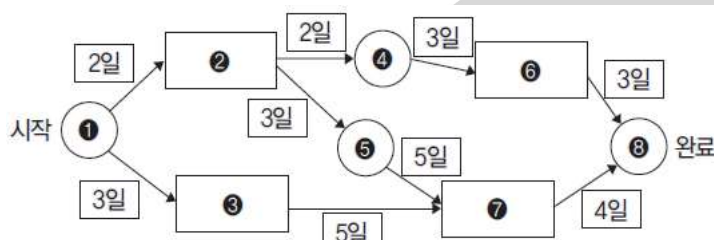
7. CPM 네트워크가 다음과 같을 때 임계경로의 소요 기일은?



- ① 10일
- ② 12일
- ③ 14일
- ④ 16일

[해설]

- 임계경로는 최장 경로를 의미합니다.
- 문제에 제시된 그림을 보고 각 경로에 대한 소요 기일을 계산한 후 가장 오래 걸린 기일을 찾으면 됩니다.



- 경로 1 : ① → ② → ④ → ⑥ → ⑧ = 2+2+3+3 = 10일
  - 경로 2 : ① → ② → ⑤ → ⑦ → ⑧ = 2+3+5+4 = 14일
  - 경로 3 : ① → ③ → ⑦ → ⑧ = 3+5+4 = 12일
- 그러므로 임계경로는 경로 2이며, 소요 기일은 14일입니다.

8. 다음과 같은 세그먼트 테이블을 가지는 시스템에서 논리 주소(2, 176)에 대한 물리 주소는?

세그먼트번호	시작주소	길이(바이트)
0	670	248
1	1752	422
2	222	198
3	996	604

- ① 398
- ② 400
- ③ 1928
- ④ 1930

[해설]

- 실기억주소는 '세그먼트의 기준번지 + 변위값'입니다.
- 논리 주소(2, 176)에서 2는 세그먼트 번호이고, 176은 변위값입니다.
- 세그먼트 번호 2의 시작 주소인 222에 변위값 176을 더하면 물리적인 실기억장치의 주소가 됩니다.

9. 4개의 페이지를 수용할 수 있는 주기억장치가 있으며, 초기에는 모두 비어 있다고 가정한다. 다음의 순서로 페이지 참조가 발생할 때, LRU 페이지 교체 알고리즘을 사용할 경우 몇 번의 페이지 결함이 발생하는가?

페이지 참조 순서 1, 2, 3, 1, 2, 4, 1, 2, 5

① 5회

② 6회

③ 7회

④ 8회

**[해설]**

4개의 페이지를 수용할 수 있는 주기억장치이므로 아래 그림과 같이 4개의 페이지 프레임으로 표현할 수 있습니다.

참조페이지	1	2	3	1	2	4	1	2	5
페이지 프레임	1	1	1	1	1	1	1	1	1
		2	2	2	2	2	2	2	2
			3	3	3	3	3	3	5
						4	4	4	4
부재 발생	●	●	●			●			●

※ ● : 페이지 부재 발생

- 참조 페이지가 페이지 테이블에 없을 경우 페이지 결함(부재)이 발생합니다.
- 초기에는 모든 페이지가 비어 있으므로 처음 1, 2, 3 페이지 적재 시 페이지 결함이 발생합니다.
- 다음 참조 페이지 1, 2는 이미 적재되어 있으므로 그냥 참조하고, 참조 페이지 4를 적재할 때 페이지 결함이 발생합니다.
- 다음 참조 페이지 1, 2 역시 이미 적재되어 있으므로 그냥 참조합니다.
- LRU 기법은 최근에 가장 오랫동안 사용되지 않은 페이지를 교체하는 기법이므로, 마지막 페이지 5를 적재할 때는 3을 제거한 후 5를 가져오게 됩니다.
- 이러한 과정으로 모든 페이지에 대한 요구를 처리하고 나면 총 페이지 결함 발생 횟수는 5회입니다.

10. 4개의 페이지를 수용할 수 있는 주기억장치가 있으며, 초기에는 모두 비어 있다고 가정한다. 다음의 순서로 페이지 참조가 발생할 때, FIFO 페이지 교체 알고리즘을 사용할 경우 페이지 결함의 발생 횟수는?

페이지 참조 순서 : 1, 2, 3, 1, 2, 4, 5, 1

① 6회

② 7회

③ 8회

④ 9회

[해설]

4개의 페이지를 수용할 수 있는 주기억장치이므로 아래 그림과 같이 4개의 페이지 프레임으로 표현할 수 있습니다.

참조페이지	1	2	3	1	2	4	5	1
페이지 프레임	1	1	1	1	1	1	5	5
		2	2	2	2	2	2	1
			3	3	3	3	3	3
						4	4	4
부재 발생	●	●	●			●	●	●

※ ● : 페이지 부재 발생

- 참조 페이지가 페이지 테이블에 없을 경우 페이지 결함(부재)이 발생합니다.
- 초기에는 모든 페이지가 비어 있으므로 처음 1, 2, 3 페이지 적재 시 페이지 결함이 발생합니다.
- 다음 참조 페이지 1, 2는 이미 적재되어 있으므로 그냥 참조하고, 4 페이지 적재 시 페이지 결함이 발생합니다.
- FIFO 기법은 가장 먼저 들어와 있었던 페이지를 교체하는 기법이므로 참조 페이지 5를 적재할 때에는 1을 제거한 후 5를 가져오게 됩니다.
- 이러한 과정으로 모든 페이지에 대한 요구를 처리하고 나면 총 페이지 결함 발생 횟수는 6회입니다.

11. 3개의 페이지 프레임을 갖는 시스템에서 페이지 참조 순서가 1, 2, 1, 0, 4, 1, 3일 경우 FIFO 알고리즘에 의한 페이지 교체의 경우 프레임의 최종 상태는?

- ① 1, 2, 0
- ② 2, 4, 3
- ③ 1, 4, 2
- ④ 4, 1, 3

**[해설]**

3개의 페이지를 수용할 수 있는 주기억장치이므로 아래 그림과 같이 3개의 페이지 프레임으로 표현할 수 있습니다.

참조페이지	1	2	1	0	4	1	3
페이지 프레임	1	1	1	1	4	4	4
		2	2	2	2	1	1
				0	0	0	3
부재 발생	●	●		●	●	●	●

※ ● : 페이지 부재 발생

- 참조 페이지가 페이지 테이블에 없을 경우 페이지 결함(부재)이 발생합니다.
- 초기에는 모든 페이지가 비어 있으므로 처음 1, 2 페이지 적재 시 페이지 결함이 발생합니다.
- 다음 참조 페이지 1은 이미 적재되어 있으므로 그냥 참조하고, 0 페이지 적재 시 페이지 결함이 발생합니다.
- FIFO 기법은 가장 먼저 들어와 있었던 페이지를 교체하는 기법이므로 참조 페이지 4를 적재할 때에는 1을 제거한 후 4를 가져오게 됩니다.
- 이러한 과정으로 모든 페이지에 대한 요구를 처리하고 나면 총 페이지 결함 발생 횟수는 6회이고 마지막 프레임의 최종 상태는 4, 1, 3입니다.

12. 평가 점수에 따른 성적부여는 다음 표와 같다. 이를 구현한 소프트웨어를 경계값 분석 기법으로 테스트하고자 할 때 다음 중 테스트 케이스의 입력값으로 옳지 않은 것은?

평가 점수	성적
80~100	A
60~79	B
0~59	C

- ① 59
- ② 80
- ③ 90
- ④ 101

**[해설]**

경계값 분석 기법은 입력 조건의 경계값을 테스트 케이스로 선정하여 검사하는 기법으로, 성적이 분리되는 평가 점수의 경계값인 101, 100, 80, 79, 60, 59, 0, -1이 적절한 입력값에 해당됩니다.



13. 다음 두 릴레이션 R1과 R2의 카티션 프로덕트(Cartesian Product) 수행 결과는?

[R1]	학년	[R2]	학과
	1		컴퓨터
	2		국문
	3		수학

①

학년	학과
1	컴퓨터
2	국문
3	수학

②

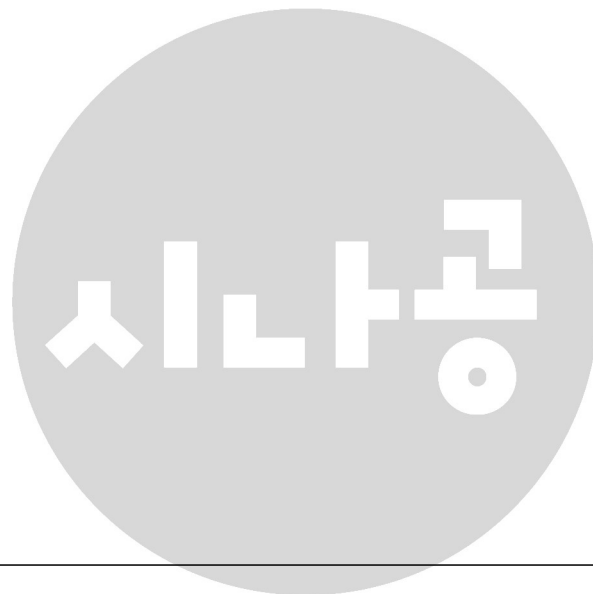
학년	학과
2	컴퓨터
2	국문
2	수학

③

학년	학과
3	컴퓨터
3	국문
3	수학

④

학년	학과
1	컴퓨터
1	국문
1	수학
2	컴퓨터
2	국문
2	수학
3	컴퓨터
3	국문
3	수학



**[해설]**

- 교차곱(Cartesian Product)은 두 릴레이션의 차수(Degree, 속성의 수)는 더하고, 카디널리티(튜플의 수)는 곱하면 됩니다.
- 차수는  $1 + 1 = 2$ 이고, 카디널리티는  $3 \times 3 = 9$ 입니다.

14. 릴레이션 R의 차수가 4이고 카디널리티가 5이며, 릴레이션 S의 차수가 6이고 카디널리티가 7일 때, 두 개의 릴레이션을 카티션 프로덕트한 결과의 새로운 릴레이션의 차수와 카디널리티는 얼마인가?

- ① 24, 35  
 ② 24, 12  
 ③ 10, 35  
 ④ 10, 12

**[해설]**

- 교차곱(Cartesian Product)은 두 릴레이션의 차수(Degree, 속성의 수)는 더하고, 카디널리티(튜플의 수)는 곱하면 됩니다.
- 차수는  $4 + 6 = 10$ 이고, 카디널리티는  $5 \times 7 = 35$ 입니다.

15. 다음 R과 S 두 릴레이션에 대한 Division 연산의 수행 결과는?

[R]

D1	D2	D3
a	1	A
b	1	A
c	2	A
d	2	B

[S]

D2	D3
1	A

①

D3
A
B

②

D2
2
2

③

D3
A

④

D1
a
b

**[해설]**

$X \supset Y$ 인 두 개의 릴레이션  $R(X)$ 와  $S(Y)$ 가 있을 때,  $R$ 의 속성이  $S$ 의 속성값을 모두 가진 튜플에서  $S$ 가 가진 속성을 제외한 속성만을 구하는 연산을 Division이라고 합니다.

① 릴레이션  $R$ 에서 릴레이션  $S$ 의 속성값을 모두 가진 튜플을 추출하면 다음과 같습니다.

D1	D2	D3
a	1	A
b	1	A

② 릴레이션  $S$ 가 가진 속성을 제외하게 되면 다음과 같습니다.

D1
a
b

16. 리눅스에서 생성된 파일 권한이 644일 경우 umask 값은?

- ① 022  
② 666  
③ 777  
④ 755

**[해설]**

- umask는 UNIX에서 파일이나 디렉터리의 초기 권한을 설정할 때 사용하는 값으로, 파일의 경우 666에서 umask를 뺀 값을, 디렉터리의 경우 777에서 umask를 뺀 값을 초기 접근 권한으로 갖습니다.

- 문제에서 파일 권한이 644라고 하였으므로, 다음과 같은 공식으로 umask의 값을 구할 수 있습니다.

$$666 - \text{umask} = 644$$

$$\text{umask} = 666 - 644 = 022$$

17. CIDR(Classless Inter-Domain Routing) 표기로 203.241.132.82/27과 같이 사용되었다면, 해당 주소의 서브넷 마스크(subnet mask)는?

- ① 255.255.255.0
- ② 255.255.255.224
- ③ 255.255.255.240
- ④ 255.255.255.248

**[해설]**

- CIDR(Classless Inter-Domain Routing)은 클래스 없는 도메인 간 라우팅 기법으로, CIDR 기법 사용 시 서브넷 마스크는 IP 주소 뒤의 숫자를 이용해 구할 수 있습니다.
- 203.241.132.82/27 네트워크의 서브넷 마스크는 1의 개수가 27개, 즉 11111111 11111111 11111111 11100000 → 255.255.255.224가 됩니다.

18. 192.168.1.0/24 네트워크를 FLSM 방식을 이용하여 4개의 Subnet으로 나누고 IP Subnet-zero를 적용했다. 이 때 Subnetting된 네트워크 중 4번째 네트워크의 4번째 사용 가능한 IP는 무엇인가?

- ① 192.168.1.192
- ② 192.168.1.195
- ③ 192.168.1.196
- ④ 192.168.1.198

**[해설]**

- 192.168.1.0/24 네트워크의 서브넷 마스크는 1의 개수가 24개, 즉 11111111 11111111 11111111 00000000 → 255.255.255.0인 C 클래스에 속하는 네트워크입니다. 이 네트워크를 4개의 Subnet으로 나눠야 하는데, Subnet을 나눌 때는 서브넷 마스크가 0인 부분, 즉 마지막 8비트를 이용해 구분할 수 있습니다. 또한 Subnet을 나눌 때 “4개의 네트워크로 나눈다”는 것처럼 네트워크가 기준일 때는 왼쪽을 기준으로 4개가 포함된 Bit 만큼을 네트워크로 할당하고 나머지 비트로 호스트를 할당하면 됩니다. 4개가 포함되는 비트는  $2^2=4$ ( $2^1$ 은 2로 4개를 포함 못함)이므로 2비트를 제외한 나머지 6비트로 호스트를 구성합니다.

네트워크ID		호스트ID					
0	0	0	0	0	0	0	0

- 호스트ID가 6Bit로 설정되었고, 문제에서 FLSM 방식을 이용한다고 했으므로 4개의 네트워크에 고정된 크기인 64개( $2^6=64$ )씩 할당하면 다음과 같습니다.

네트워크	호스트 수	IP 주소 범위
1	32	192.168.1.0 ~ 63
2	32	192.168.1.64 ~ 127
3	32	192.168.1.128 ~ 191
4	32	192.168.1.192 ~ 255

- 4번째 네트워크의 시작 주소인 192.168.1.192는 네트워크의 대표 주소로 사용되므로 사용 가능한 주소는 193부터 4번째에 해당하는 주소는 192.168.1.196입니다.
- ※ ip subnet-zero를 적용했다는 것은 Subnet 부분이 모두 0인 192.168.1.0은 사용하지 않았는데, IP 주소가 부족해지면서 Subnet 부분이 모두 0인 주소도 IP 주소로 사용할 수 있도록 한다는 의미입니다.

19. 200.1.1.0/24 네트워크를 FLSM 방식을 이용하여 10개의 Subnet으로 나누고, ip subnet-zero를 적용했다. 이때 서브네팅된 네트워크 중 10번째 네트워크의 broadcast IP 주소는?

- ① 200.1.1.159
- ② 201.1.5.175
- ③ 202.1.11.254
- ④ 203.1.255.245

**[해설]**

- 200.1.1.0/24 네트워크의 서브넷 마스크는 1의 개수가 24개, 즉 11111111 11111111 11111111 00000000 → 255.255.255.0인 C 클래스에 속하는 네트워크입니다. 이 네트워크를 10개의 Subnet으로 나눠야 하는데, Subnet을 나눌 때는 서브넷 마스크가 0인 부분, 즉 마지막 8비트를 이용해 나눠야 합니다. 또한 “10개의 Subnet으로 나눈다”는 것처럼 네트워크가 기준일 때는 왼쪽을 기준으로 10개가 포함된 Bit 만큼을 네트워크로 할당하고, 나머지 비트로 호스트를 구성하면 됩니다. 10개가 포함되는 비트는  $2^4 = 16$ ( $2^3$ 은 8로 10개를 포함 못함)이므로 4비트를 제외한 나머지 4비트로 호스트를 구성합니다.

네트워크ID				호스트ID			
0	0	0	0	0	0	0	0

- 호스트ID가 4Bit로 설정되었고, 문제에서 FLSM 방식을 이용한다고 했으므로 10개의 네트워크에 고정된 크기인 16개( $2^4=16$ )씩 할당하면 다음과 같습니다.

네트워크	호스트 수	IP 주소 범위
1	16	200.1.1.0 ~ 200.1.1.15
2	16	200.1.1.16 ~ 200.1.1.31
3	16	200.1.1.32 ~ 200.1.1.47
4	16	200.1.1.48 ~ 200.1.1.63
5	16	200.1.1.64 ~ 200.1.1.79
6	16	200.1.1.80 ~ 200.1.1.95
7	16	200.1.1.96 ~ 200.1.1.111
8	16	200.1.1.112 ~ 200.1.1.127
9	16	200.1.1.128 ~ 200.1.1.143
10	16	200.1.1.144 ~ 200.1.1.159

- ※ ‘subnet-zero’는 Subnet 부분이 모두 0인 네트워크를 의미하며 일반적으로 사용하지 않는데, IP 주소가 부족할 경우 ‘ip subnet-zero’를 적용하여 이 부분도 IP 주소로 사용할 수 있도록 합니다.
- ※ broadcast 주소는 해당 IP 주소 범위에서 가장 마지막 주소를 의미합니다.

20. 메모리 관리 기법 중 Worst Fit 방법을 사용할 경우 10K 크기의 프로그램 실행을 위해서는 어느 부분에 할당되는가?

영역 번호	메모리 크기	사용 여부
NO.1	8K	FREE
NO.2	12K	FREE
NO.3	10K	IN USE
NO.4	20K	IN USE
NO.5	16K	FREE

① NO.2

② NO.3

③ NO.4

④ NO.5

[해설]

- 최악 적합(Worst-Fit)은 데이터가 들어갈 수 있는 크기의 빈 영역 중에서 단편화를 가장 많이 남기는 분할 영역에 배치시키는 방법입니다.
- 사용 여부가 FREE인 메모리 중 가장 메모리 크기가 큰 영역인 NO.5에 배치됩니다.

21. 빈 기억공간의 크기가 20KB, 16KB, 8KB, 40KB 일 때 기억장치 배치 전략으로 "Best Fit"을 사용하여 17KB의 프로그램을 적재할 경우 내부 단편화의 크기는 얼마인가?

① 3KB

② 23KB

③ 64KB

④ 67KB

[해설]

- 최적 적합(Best-Fit)은 데이터가 들어갈 수 있는 크기의 빈 영역 중 단편화를 가장 적게 남기는 분할 영역에 배치시키는 방법으로, 17KB보다 큰 영역 중 가장 작은 영역인 20KB에 배치됩니다.
- 내부 단편화는 분할된 영역이 할당될 프로그램의 크기보다 크기 때문에 프로그램이 할당된 후 사용되지 않고 남아 있는 빈 공간을 의미하므로,  $20\text{KB} - 17\text{KB} = 3\text{KB}$ 입니다.

22. 사용자가 요청한 디스크 입·출력 내용이 다음과 같은 순서로 큐에 들어 있을 때 SSTF 스케줄링을 사용한 경우의 처리 순서는? (단, 현재 헤드 위치는 53이고, 제일 안쪽이 1번, 바깥쪽이 200번 트랙이다.)

큐의 내용 : 98 183 37 122 14 124 65 67

① 53 → 65 → 67 → 37 → 14 → 98 → 122 → 124 → 183

② 53 → 98 → 183 → 37 → 122 → 14 → 124 → 65 → 67

③ 53 → 37 → 14 → 65 → 67 → 98 → 122 → 124 → 183

④ 53 → 67 → 65 → 124 → 14 → 122 → 37 → 183 → 98

[해설]

현재 헤드는 53트랙에 있으며, SSTF는 현재 위치에서 가장 가까운 거리에 있는 트랙의 요청을 먼저 서비스하므로 이동 순서는 '53 → 65 → 67 → 37 → 14 → 98 → 122 → 124 → 183'이고, 총 이동 거리는 ' $12 + 2 + 30 + 23 + 84 + 24 + 2 + 59 = 236$ '입니다.

23. 두 명의 개발자가 5개월에 걸쳐 10000 라인의 코드를 개발하였을 때, 월별(man-month) 생산성 측정을 위한 계산 방식으로 가장 적합한 것은?

- ① 10000/2
- ② 10000/(5×2)
- ③ 10000/5
- ④ (2×10000)/5

**[해설]**

- 노력(인월) = 개발 기간 × 투입 인원 = LOC / 1인당 월평균 생산 코드 라인 수
- 생산성 = LOC / 노력(인월)
- 10000 / (5×2) = 1000

24. LOC 기법에 의하여 예측된 총 라인수가 36000라인, 개발에 참여할 프로그래머가 6명, 프로그래머들의 평균 생산성이 월간 300라인일 때 개발에 소요되는 기간을 계산한 결과로 가장 옳은 것은?

- ① 5개월
- ② 10개월
- ③ 15개월
- ④ 20개월

**[해설]**

- 노력(인월) = 개발 기간 × 투입 인원 = LOC / 1인당 월평균 생산 코드 라인 수
- 개발 기간 = 노력(인월) / 투입 인원
- 총 라인수가 36000이고, 프로그래머들의 평균 생산성이 월간 300라인이므로 노력(인월)은 120입니다.
- 노력(인월)이 120이고, 프로그래머가 6명이므로 개발 기간은 20개월입니다.

25. 다음과 같은 프로세스가 차례로 큐에 도착하였을 때, SJF(Shortest Job First) 정책을 사용할 경우 가장 먼저 처리되는 작업은?

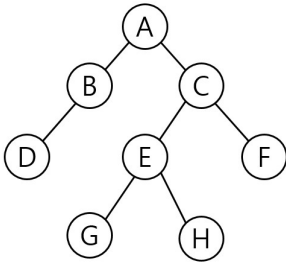
프로세스 번호	실행시간
P1	6
P2	8
P3	4
P4	3

- ① P1
- ② P2
- ③ P3
- ④ P4

**[해설]**

- SJF는 준비상태 큐에서 기다리고 있는 프로세스들 중에서 실행 시간이 가장 짧은 프로세스에게 먼저 CPU를 할당하는 기법입니다.
- 가장 짧은 실행시간을 갖고 있는 P4가 첫 번째로 수행됩니다.

26. 다음 트리의 차수(Degree)와 단말 노드(Terminal Node)의 수는?

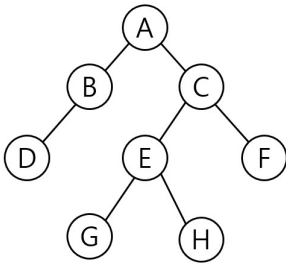


- ① 차수 : 4, 단말 노드 : 4
- ② 차수 : 2, 단말 노드 : 4
- ③ 차수 : 4, 단말 노드 : 8
- ④ 차수 : 2, 단말 노드 : 8

**[해설]**

- 트리의 차수(Degree)는 가장 차수가 많은 노드의 차수이고, 단말 노드(Terminal Node)는 자식이 하나도 없는 노드입니다.
- A, C, E의 차수 2가 차수 중 가장 높으므로 트리의 차수는 2가 되고, 자식이 하나도 없는 노드는 D, G, H, F로 총 4개가 됩니다.

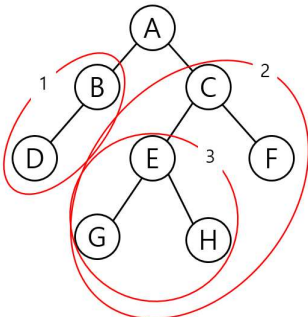
27. 다음 트리를 Preorder 운행법으로 운행할 경우 다섯 번째로 탐색되는 것은?



- ① C
- ② E
- ③ G
- ④ H

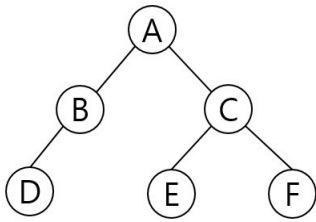
**[해설]**

먼저 서브트리를 하나의 노드로 생각할 수 있도록 서브트리 단위로 묶습니다.



- ① Preorder는 Root → Left → Right이므로 A12입니다.
  - ② 1은 BD이므로 ABD2입니다.
  - ③ 2는 C3F이므로 ABDC3F입니다.
  - ④ 3은 EGH이므로 ABDCEGHF입니다.
- 따라서 다섯 번째로 탐색되는 노드는 E가 됩니다.

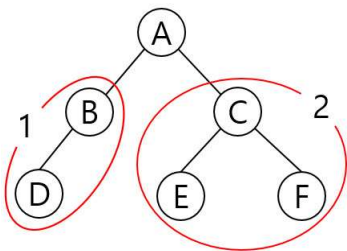
28. 다음 트리를 후위 순회(Post Traversal)한 결과는?



- ① A B D C E F
- ② D B A E C F
- ③ A B C D E F
- ④ D B E F C A

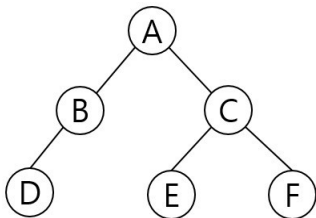
[해설]

먼저 서브트리를 하나의 노드로 생각할 수 있도록 서브트리 단위로 묶습니다.



- ❶ Postorder는 Left → Right → Root이므로 12A가 됩니다.
- ❷ 1은 DB이므로 DB2A가 됩니다.
- ❸ 2는 EFC이므로 DBEFCA가 됩니다.

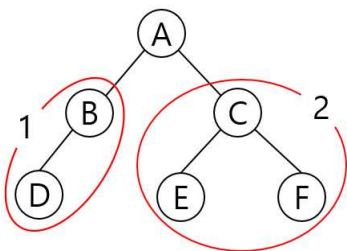
29. 다음 트리에 대한 중위 순회 운행 결과는?



- ① ABDCEF
- ② ABCDEF
- ③ DBECFA
- ④ DBAECF

[해설]

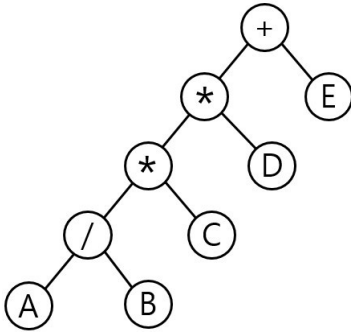
먼저 서브트리를 하나의 노드로 생각할 수 있도록 서브트리 단위로 묶습니다.



- ❶ 중위 순회(Inorder)는 Left → Root → Right이므로 1A2가 됩니다.
- ❷ 1은 DB이므로 DBA2가 됩니다.
- ❸ 2는 ECF이므로 DBAECF가 됩니다.



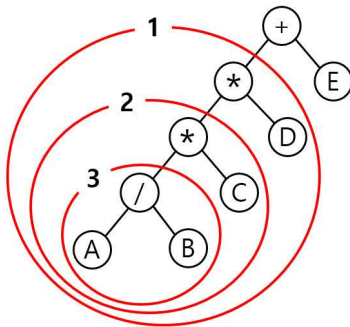
30. 다음 트리를 전위 순회(Preorder Traversal)한 결과는?



- ① + \* A B / \* C D E
- ② A B / C \* D \* E +
- ③ A / B \* C \* D + E
- ④ + \* \* / A B C D E

[해설]

먼저 서브 트리를 하나의 노드로 생각할 수 있도록 서브 트리 단위로 묶습니다.



- ① Preorder는 Root → Left → Right이므로 +1E입니다.
- ② 1은 \*2D이므로 ++2DE입니다.
- ③ 2는 \*3C이므로 \*\*\*3CDE입니다.
- ④ 3은 /AB이므로 +\*\*/ABCDE입니다.

31. n개의 노드로 구성된 무방향 그래프의 최대 간선수는?

- ①  $n - 1$
- ②  $n/2$
- ③  $n(n - 1)/2$
- ④  $n(n + 1)$

[해설]

- 무방향 그래프 최대 간선 수 :  $n(n-1)/2$
- 방향 그래프 최대 간선 수 :  $n(n-1)$

32. 다음 전위식(prefix)을 후위식(postfix)으로 옮겨 표현한 것은?

- / \* A + B C D E

- ① A B C + D / \* E -
- ② A B \* C D / + E -
- ③ A B \* C + D / E -
- ④ A B C + \* D / E -

[해설]

전위식(Prefix)은 연산자를 해당 피연산자 두 개의 앞(왼쪽)으로 이동시킨 것입니다. 그러므로 연산자와 피연자 두 개를 묶은 후 연산자를 피연산자 두 개의 뒤(오른쪽)로 옮겨 놓으면 후위식(Postfix)이 됩니다.

① 인접한 피연산자 두 개와 왼쪽의 연산자를 괄호로 묶는다.

( - ( / ( \* A ( + B C ) ) D ) E )

② 연산자를 피연산자의 뒤로 옮긴다.

( - ( / ( \* A ( + B C ) ) D ) E )

③ 필요없는 괄호를 제거한다.

A B C + \* D / E -

33. 다음 Postfix로 표현된 연산식의 연산 결과로 옳은 것은?

3 4 \* 5 6 \* +

- ① 35
- ② 42
- ③ 81
- ④ 360

[해설]

Postfix(후위 표기 방식) 연산식은 연산자가 해당 피연산자 두 개의 뒤(오른쪽)에 놓인 것을 말합니다. 그러므로 피연자 2개와 연산자를 묶은 후 연산자를 피연산자 사이에 옮겨 놓고 계산하면 됩니다.

① 피연산자 2개와 오른쪽으로 인접한 연산자 1개를 묶습니다.

3 4 \* 5 6 \* +

↓

(( 3 4 \* ) ( 5 6 \* ) + )

② 연산자를 피연산자 사이로 옮깁니다.

(( 3 4 \* ) ( 5 6 \* ) + )

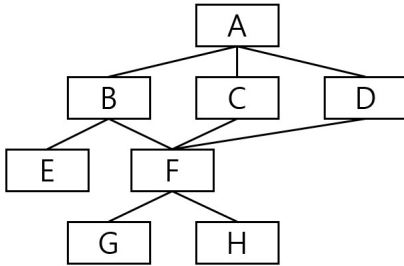
↓

(( 3 \* 4 ) + ( 5 \* 6 ) )

③ 연산을 수행합니다.

(( 3 \* 4 ) + ( 5 \* 6 ) ) = 12 + 30 = 42

34. 다음은 어떤 프로그램 구조를 나타낸다. 모듈 F에서의 fan-in과 fan-out의 수는 얼마인가?



- ① fan-in : 2, fan-out : 3
- ② fan-in : 3, fan-out : 2
- ③ fan-in : 1, fan-out : 2
- ④ fan-in : 2, fan-out : 1

**[해설]**

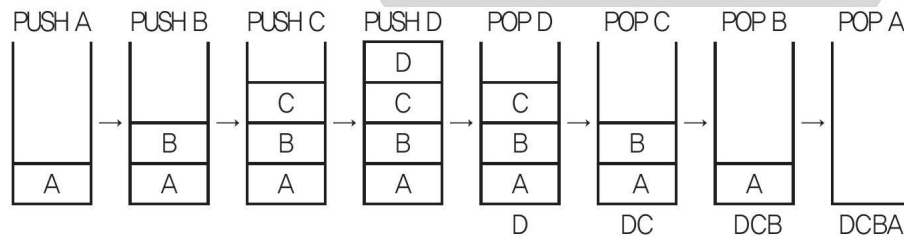
- 모듈에 들어오면(in) 팬인(fan-in), 모듈에서 나가면(out) 팬아웃(fan-out)입니다.
- F에 들어오는 선은 3개, 나가는 선은 2개이므로, 팬인과 팬아웃은 각각 3과 2입니다.

35. 순서가 A, B, C, D로 정해진 입력 자료를 스택에 입력한 후 출력한 결과로 불가능한 것은?

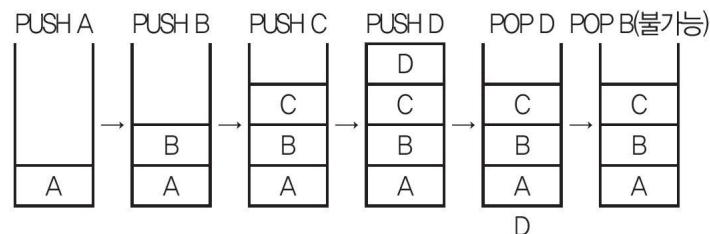
- ① D, C, B, A
- ② B, C, D, A
- ③ C, B, A, D
- ④ D, B, C, A

**[해설]**

이 문제는 문제의 자료가 각 보기의 순서대로 출력되는지 스택을 이용해 직접 입·출력을 수행해 보면 됩니다. PUSH는 스택에 자료를 입력하는 명령이고, POP는 스택에서 자료를 출력하는 명령입니다. 먼저 ①번은 다음과 같은 순서로 작업하면 모두 출력할 수 있습니다.



④번은 'D'를 출력한 후 'B'를 출력해야 하는데, 'C'를 출력하지 않고는 'B'를 출력할 수 없으므로 불가능합니다.



36. HRN 방식으로 스케줄링 할 경우, 입력된 작업이 다음과 같을 때 처리되는 작업 순서로 옳은 것은?

작업	대기 시간	서비스(실행) 시간
A	5	20
B	40	20
C	15	45
D	20	2

- ① A → B → C → D  
 ② A → C → B → D  
 ③ D → B → C → A  
 ④ D → A → B → C

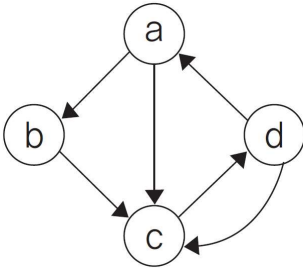
[해설]

HRN 기법의 우선순위 공식은 '(대기 시간 + 서비스 시간) / (서비스 시간)'입니다.

- A 작업 :  $(5 + 20) / 20 = 1.25$
- B 작업 :  $(40 + 20) / 20 = 3$
- C 작업 :  $(15 + 45) / 45 = 1.33$
- D 작업 :  $(20 + 2) / 2 = 11$

계산된 숫자가 클수록 우선순위가 높습니다.

37. 제어 흐름 그래프가 다음과 같을 때 McCabe의 cyclomatic 수는 얼마인가?



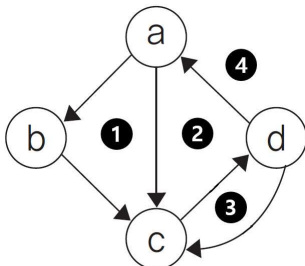
- ① 3  
 ② 4  
 ③ 5  
 ④ 6

[해설]

제어 흐름도에서 순환 복잡도(cyclomatic)는 다음과 같이 2가지 방법으로 계산할 수 있습니다.

① 영역 수를 계산합니다.

내부 영역 3(①, ②, ③) + 외부 영역 1(④) = 4



②  $V(G) = E - N + 2$  (E는 화살표 수, N은 노드 수)  
 $= 6 - 4 + 2 = 4$

38. 다음 두 릴레이션에서 외래키로 사용된 것은? (단, 밑줄 친 속성은 기본키이다.)

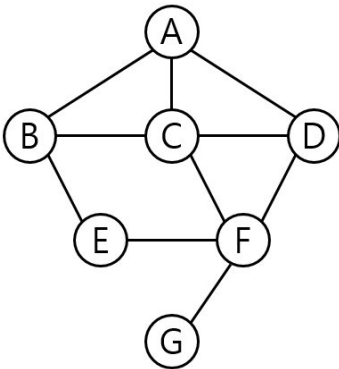
과목(과목번호, 과목명)
수강(수강번호, 학번, <u>과목번호</u> , 학기)

- ① 수강번호
- ② 과목번호
- ③ 학번
- ④ 과목명

[해설]

두 릴레이션에 공통으로 존재하는 속성명은 '과목번호'입니다. <과목> 릴레이션의 '과목번호'는 기본키 속성으로 동일한 속성값이 존재할 수 없고, <수강> 릴레이션의 '과목번호'는 일반 속성으로 여러 속성값이 존재할 수 있으므로 <수강> 릴레이션의 '과목번호' 속성이 <과목> 릴레이션의 기본키 속성 '과목번호'를 참조하는 외래키가 됩니다.

39. 다음 그래프에서 정점 A를 선택하여 깊이 우선 탐색(DFS)으로 운행한 결과는?



- ① ABECDFG
- ② ABECFDG
- ③ ABCDEFG
- ④ ABEFGCD

[해설]

깊이 우선 탐색(DFS)은 정점에서 자식 노드 방향으로 운행하면서 형제 노드와 자식 노드가 있을 때 자식 노드를 우선 탐색하는 기법입니다. 자식 노드의 탐색이 모두 끝나면 다시 형제 노드부터 탐색을 시작하는 방식으로, 모든 노드를 한 번씩 방문합니다.

- ❶ A 노드에서는 B, C, D의 세 가지의 동등한 순위의 선택지가 있습니다. D로 진행해도 올바른 운행이지만 보기의 문항이 B로만 진행되고 있으므로, B로 진행합니다. → AB
- ❷ B 노드에서는 C, E의 선택지가 있으나 자식 노드가 우선시 되므로 E로 진행합니다. → ABE
- ❸ E 노드에서는 F 외에는 선택지가 없으므로 F로 진행합니다. → ABEF
- ❹ F 노드에서는 G 외에는 선택지가 없으므로 G로 진행합니다. → ABEFG
- ❺ ❷번에서 우선순위가 밀렸던 형제 노드 C로 진행합니다. → ABEFGC
- ❻ C 노드에서는 D, F의 선택지가 있으나 우선해야 하는 자식 노드 F는 ❸번에서 이미 방문했으므로 형제 노드인 D로 진행합니다. → ABEFGCD
- ❼ 모든 노드를 한 번씩 방문했으므로 운행을 종료합니다.

40. 다음 관계형 데이터 모델에 대한 설명으로 옳은 것은?

고객ID	고객이름	거주도시
S1	홍길동	서울
S2	이정재	인천
S3	신보라	인천
S4	김흥국	서울
S5	도요새	용인

- ① Relation 3개, Attribute 3개, Tuple 5개
- ② Relation 3개, Attribute 5개, Tuple 3개
- ③ Relation 1개, Attribute 5개, Tuple 3개
- ④ Relation 1개, Attribute 3개, Tuple 5개

[해설]

문제에 제시된 표는 한 개의 릴레이션(Relation), 3개의 속성(Attribute), 5개의 튜플(Tuple)을 표현하고 있습니다.

41. 다음과 같이 레코드가 구성되어 있을 때, 이진 검색 방법으로 14를 찾을 경우 비교되는 횟수는?

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----

- ① 2
- ② 3
- ③ 4
- ④ 5

[해설]

- ① 첫 번째 값(F)과 마지막 값(L)을 이용하여 중간 값 M을 구한 후 찾으려는 값과 비교합니다.  
 $M = (1+15) / 2 = 8$   
 8이 찾으려는 값인지 확인합니다. 8은 찾으려는 값 14보다 작으므로 찾는 값은 9~15에 있습니다.  
 ← 1회 비교
- ②  $F = 9, L = 15, M = (9+15) / 2 = 12$   
 12가 찾으려는 값인지 확인합니다. 12는 찾으려는 값 14보다 작으므로 찾는 값은 13~15에 있습니다. ← 2회 비교
- ③  $F = 13, L = 15, M = (13+15) / 2 = 14$   
 14가 찾으려는 값인지 비교합니다. 14는 찾는 값입니다. ← 3회 비교