

2과목	자 료 구 조	(36~60)
출제위원 : 방송대 정광식		
출제범위 : 교재 전체 (해당 멀티미디어강의 포함)		

36. 아래의 문장을 수식으로 표현한 것으로 알맞은 것은? (4점)

정보(Information)는 자료(Data)를 처리(Process)해서 얻어진 결과(Result)다.

- ① $R = P(D)$, (R : 결과, D : 자료, P : 처리)
- ② $I = P(R)$, (I : 정보, R : 결과, P : 처리)
- ③ $P = R(D)$, (R : 결과, D : 자료, P : 처리)
- ④ $I = P(D)$, (I : 정보, D : 자료, P : 처리)

37. 현실 세계에서 관찰이나 측정을 통해서 수집된 값이나 사실을 무엇이라 하는가? (4점)

- ① 자료
- ② 정보
- ③ 자료구조
- ④ 추상화

38. 다음 설명 중 틀린 것은 무엇인가? (3점)

- ① 배열은 인덱스와 원소값의 쌍으로 구성된다.
- ② 배열의 순서는 원소값이 저장되는 물리적인 위치와 아주 밀접한 상관이 있다.
- ③ 배열의 인덱스값을 이용해서 원소값에 직접 접근한다.
- ④ 배열의 각 원소값의 의미적인 순서는 인덱스의 순서와 일치한다.

39. 다음 설명 중 틀린 것은 무엇인가? (4점)

- ① 스택은 자료의 삽입과 삭제가 같은 변수를 통해 제어된다.
- ② 스택은 객체와 객체가 저장되는 순서를 기억하는 방법에 관한 추상자료형이다.
- ③ 스택의 크기는 가변적이다.
- ④ 후위 표기식은 연산자를 피연산자의 뒤에 표기하는 방법이다.

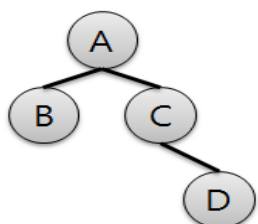
40. 다음 큐에 대한 설명으로 틀린 것은 무엇인가? (3점)

- ① 원형 큐의 입구(rear 변수)와 출구(front 변수)를 연결하여 데이터 공간을 연속적으로 사용하기 위해 제안되었다.
- ② 큐는 서로 다른 부분에서 삽입과 삭제가 발생하는 FIFO 특성을 갖는 순서 리스트이다.
- ③ 큐가 가득 차는 경우는 삽입되는 부분의 rear 변수와 삭제되는 부분의 front 변수를 이용하여 찾아낸다.
- ④ 삽입되는 부분의 rear 변수가 마지막을 가리키면, 큐에 포함된 원소의 갯수는 큐의 크기와 같고 큐가 가득 찬 경우이다.

41. 단순 연결리스트의 사용되지 않는 마지막 노드의 링크 부분을 활용하기 위해 제안된 리스트는 무엇인가? (3점)

- ① 원형 연결 리스트
- ② 이중 연결 리스트
- ③ 복합 연결 리스트
- ④ 순서 연결 리스트

42. 다음 이진 트리의 후위순회 결과는 무엇인가? (3점)



- ① ABCD
- ② CABD
- ③ DBAC
- ④ BCDA

43. 다음 설명 중 틀린 것은 무엇인가? (4점)

- ① 트리는 논리적 계층을 표현하는 자료구조이다.
- ② 트리에서 루트는 부모가 없는 노드이다.
- ③ 트리에서 각 노드의 차수는 진출차수이다.
- ④ 이진트리는 모든 노드의 차수가 2인 트리이다.

44. 널 링크를 이용하여 이진 트리 순회를 편리하고 빠르게 향상시킨 트리는 무엇인가? (3점)

- ① 경사 이진트리
- ② 스레드 이진트리
- ③ 포화 이진트리
- ④ 균형 이진트리

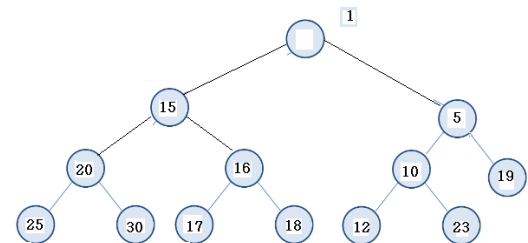
45. 다음 전위 순회 스레드 트리의 전위 순회에서 빈 칸 [가]에 알맞은 것은 무엇인가? (2점)

```

① void Preordthr(struct TFNode *root) {
    struct TFNode *p ;
    p = root ;
    while(p != null) {
        printf("%d", p->info) ;
        [ 가 ]
    }
}
    
```

- ① $p = p \rightarrow \text{left} ;$
- ② $p = p \rightarrow \text{right} ;$
- ③ $p = p \rightarrow \text{left} \rightarrow \text{right} ;$
- ④ $p = p \rightarrow \text{right} \rightarrow \text{left} ;$

46. 다음 힙에서 루트(노드 값 1)를 삭제한 결과는 무엇인가? (2점)



- ①
- ②
- ③
- ④

47. 선택트리에 대한 설명으로 틀린 것은 무엇인가? (3점)

- ① 선택트리에서는 승자트리와 패자트리가 있다.
- ② 패자트리는 루트 노드 위에 최상위 0번 노드를 갖는다.
- ③ 승자트리는 루트 노드 위에 최상위 0번 노드를 가지며, 각 노드가 두 자식노드보다 더 작은 값을 갖는다.
- ④ 선택 트리를 사용하면 정렬을 위한 합병 횟수를 줄일 수 있다.

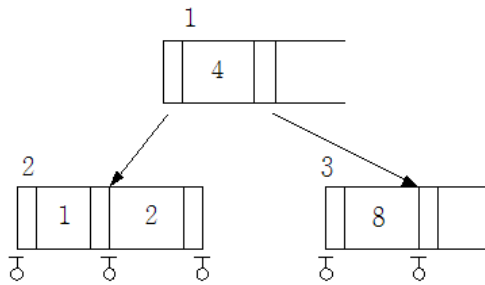
48. 자주 탐색하는 키를 가진 노드를 루트에 가깝게 위치하는 트리는 무엇인가? (2점)

- ① Splay 트리
- ② BS 트리
- ③ BB 트리
- ④ AVL 트리

49. 차수가 m인 B 트리의 조건으로 틀린 것은 무엇인가? (3점)

- ① 루트와 단말 노드를 제외한 트리의 각 노드는 최소 $\lceil m/2 \rceil$ 개의 서브트리를 갖는다.
- ② 트리의 모든 단말 노드는 같은 레벨에 있다.
- ③ 트리의 루트는 최소한 2개의 서브트리를 갖는다.
- ④ 잎 노드를 순차적으로 연결하는 포인터 집합이 있다.

50. 다음과 같은 트리를 무엇인가? (3점)



- ① 이진 트리
- ② 1-2 트리
- ③ 2-3 트리
- ④ 2-3-4 트리

51. 그래프에 대한 설명으로 틀린 것은 무엇인가? (2점)

- ① 두 정점이 사이클로 연결되었을 때, 두 정점이 인접한다고 정의한다.
- ② 사이클이 없는 그래프를 트리 혹은 무사이클 그래프라 한다.
- ③ 한 정점에서 출발하여 자신으로 연결하는 간선을 루프라고 한다.
- ④ 시작점과 끝점이 같은 경로를 사이클이라고 한다.

52. 그래프 G의 모든 정점과 간선의 일부를 포함하는 트리는 무엇인가? (3점)

- ① m원 트리
- ② 균형 트리
- ③ 신장 트리
- ④ 이진 탐색 트리

53. 다음 중 m원 탐색 트리에 대한 설명으로 틀린 것은 무엇인가? (p_0, p_1, \dots, p_n 은 서브트리에 대한 포인터이고 k_0, \dots, k_{n-1} 은 키값이다. 또한, $n \leq m-1$ 이 성립한다.) (2점)

- ① 노드 v_i 의 키를 k_i 라 할 때, v_i 의 왼쪽 서브트리에 있는 모든 노드의 키값은 v_i 의 키값보다 작다.
- ② $i = 0, \dots, n-2$ 인 i 에 대해 $k_i < k_{i+1}$ 를 만족한다.
- ③ $i = 0, \dots, n-1$ 인 i 에 대해 p_i 가 가리키는 서브트리의 모든 키값은 k_i 의 키값보다 작다.
- ④ p_n 이 가리키는 서브트리의 모든 키값은 k_{n-1} 의 키값보다 크다.

54. 다음 중 B 트리에 대한 설명으로 틀린 것은 무엇인가? (3점)

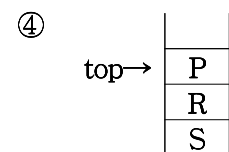
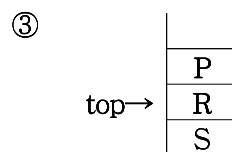
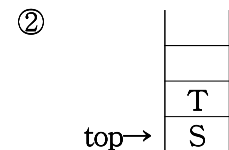
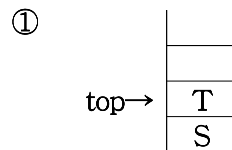
- ① 루트와 단말 노드를 제외한 트리의 각 노드는 최소 $\lceil m/2 \rceil$ 개의 서브트리를 갖는다.
- ② 트리의 루트는 최소한 2개의 서브트리를 갖는다.
- ③ 트리의 모든 단말 노드는 같은 레벨에 있다.
- ④ 루트가 가장 작은 값을 갖고 부모는 자식보다 작은 값을 갖는다.

55. 다수의 간선을 선택하고 그래프의 모든 정점들로 구성된 숲에서 시작하여 최소 비용 신장 트리를 결정하는 방법은 무엇인가? (2점)

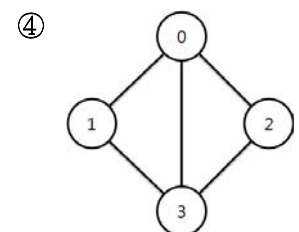
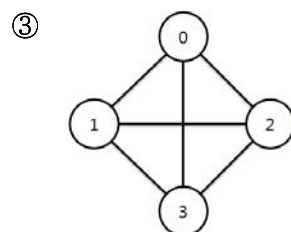
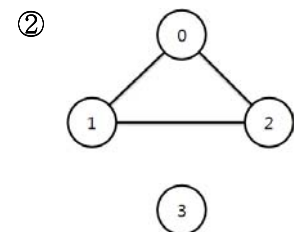
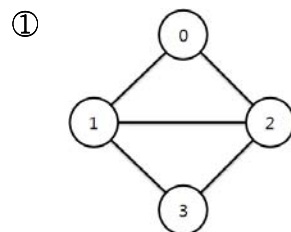
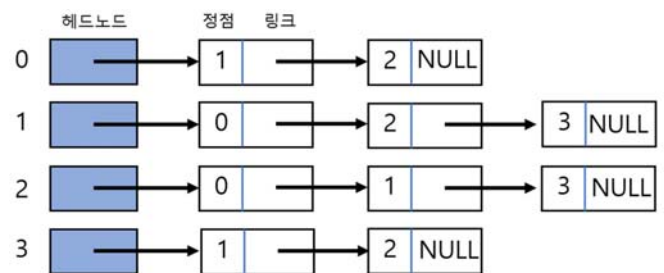
- ① Prim 방법
- ② DFS 방법
- ③ Kruscal 방법
- ④ BFS 방법

56. 다음은 스택의 연산들이다. ⑥을 수행한 결과는 무엇인가? (4점)

- ① CreateS(3);
- ② Push(stack, 'S');
- ③ Push(stack, 'T');
- ④ Pop(stack);
- ⑤ Push(stack, 'R');
- ⑥ Push(stack, 'P');
- ⑦ Push(stack, 'Q');
- ⑧ Pop(stack);

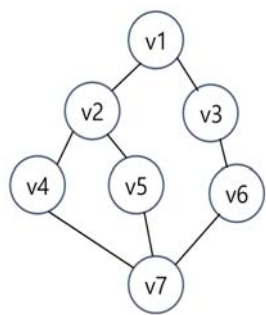


57. 다음은 G1에 대한 인접 리스트를 표현한 것이다. G1에 대한 그래프로 알맞은 것은? (2점)



[58~60]

[그림 1]



58. 다음은 깊이 우선 탐색(DFS) 알고리즘의 의사코드이며 해당 코드는 순환 호출을 이용하는 경우이다. 이때 [그림 1]의 v7 노드에 방문하기 위해선 몇 번의 ㉗ 코드가 실행되는가? (2점)

```
㉑ void DFS(int v) {
㉒     int w;
㉓     extern int VISITED[];
㉔     VISITED[v] = 1;
㉕     while(v에 인접한 모든 노드 w)
㉖         if(!VISITED[w])
㉗             DFS(w);
㉘ }
```

- ㉑ 1
- ㉒ 2
- ㉓ 3
- ㉔ 4

59. 다음은 큐를 이용한 너비 우선 탐색(BFS) 알고리즘의 의사코드이다. 아래의 코드에서 빈 칸 ㉑, ㉒에 들어갈 알맞은 코드는 무엇인가? (2점)

```
㉑ void BFS(int v) {
㉒     int w;
㉓     extern struct queue *q;
㉔     VISITED[v] = 1;
㉕     InitializeQueue(q);
㉖     AddQueue(q,v)
㉗     while( ㉑ ){
㉘         v=DeleteQueue(q);
㉙         while( ㉒ ){
㉚             if(!VISITED[w]){
㉛                 AddQueue(q,w);
㉜                 VISITED[w]=1;
㉝             }
㉞         }
㉟     }
㊱ }
```

- ㉑ ㉑ : v에 인접한 모든 노드 w
㉒ ㉒ : !q_empty()
- ㉑ ㉑ : !q_empty()
㉒ ㉒ : v에 인접한 모든 노드 w
- ㉑ ㉑ : q_empty()
㉒ ㉒ : v에 인접한 모든 노드 w
- ㉑ ㉑ : !VISITED[w]
㉒ ㉒ : v에 인접한 모든 노드 w

60. 위의 [그림 1]을 너비 우선 탐색(BFS) 알고리즘으로 탐색할 때의 노드 방문 순서로 옳은 결과는 무엇인가? (2점)

- ㉑ v1, v2, v4, v5, v3, v6, v7
- ㉒ v1, v2, v3, v6, v5, v4, v7
- ㉓ v1, v2, v3, v4, v5, v6, v7
- ㉔ v1, v2, v4, v7, v3, v5, v6