1. 해싱의 성능을 결정하는 요소로서 전체 파일공간 크기에 대한 실제 저장되는 레코드가 차지하는 크기의 비율을 ( )이라고 하며 이것이 높아질수록 충돌 확률은 (높아/낮아) 진다.
2. 직접 파일 구현 시 오버플로우 해결 기법은 탐색 Bucket 주소를 동적으로 계산하는 ( )과 overflow record를 별도의 구역에 저장하고 home bucket에 chain으로 연결하는 ( )이 있다.
3. 트리구조의 성능의 척도는 트리의 ( )이다. 따라서 트리 구성 시 이것을 (최소화/최대화)하는 것이 바람직하며 이를 위하여 AVL, B트리 등은 삽입 삭제 시 트리의 ( )을 위한 절차를 거친다.
4. AVL트리 등 탐색트리에서 순차탐색을 위한 순회법은 (전위/중위/후위) 순회이다.
5. 5. 다중키 파일구조인 역화일 구조는 인덱스를 이용하는 구조로서 ( )이란 인덱스와 데이타 레코드 화일을 연결하는 도구이다.
6. 같은 문자가 무수히 많이 반복적으로 나타날 때 반복회수를 기록하여 데이터를 압축하는 방식을 ( )라고 하며, 대용량의 데이터가 순서대로 나타나는 파일에서 데이터의 변화량만 기록하는 압축하는 방식을 ( )이라고 한다.

7. 직접화일에 대한 오버플로우 해결기법 중 체이닝에 관한 설명 중 잘못된 것은?

   ① 폐쇄주소법이라고도 한다.

   ② 홈버켓이 아닌 다른 버켓에 저장되는 전치현상이 있다.

   ③ 별도의 오버플로우 구역을 사용한다.

   ④ 충돌레코드(동의어)끼리 연결되어 검색이 용이하다.

8. 다음 중 인덱스를 위한 트리 구조에 대한 설명 중 잘못된 것은?

   ① 트리의 차수, 즉, 자식 노드의 개수가 커질수록 높이가 낮아진다.

   ② 균형된 트리는 불균형 트리보다 높이가 낮다.

   ③ B-Tree 중 차수가 2인 트리는 AVL-Tree라고 할 수 있다.

   ④ 탐색트리의 노드의 킷값은 자식노드 탐색에 대한 기준값으로 사용된다.

9. 인덱스 화일에 관한 설명 중 틀린 것은?

1) 직접접근과 순차접근 모두 용이한 구조이다.

2) 인덱스는 파일의 레코드에 대한 포인터 역할을 한다.

3) 정적인덱스 방식은 반드시 오버플로우 구역을 필요로 한다.

4) 동적 인덱스 방식은 하드웨어 의존적인 설계기법이다.

10. 동적인덱스화일에 대한 설명 중 틀린 것은?

1) 인덱스블럭과 데이터블럭으로 구성되어 있다.

2) 인덱스에는 킷값과 레코드 저장주소가 기록되어진다.

3) 데이터블럭은 순차검색을 위하여 저장장치에 순차적으로 저장한다.

4) 블록에는 자유공간이 발생해 차후 삽입 시 용이하게 이용할 수 있다.

11. 다음과 같이 키와 해싱함수에 의하여 생성된 주소들의 대응관계를 생각해보자

키값	A	B	C	D
홈주소	3	3	2	2

A-B-C-D의 순서로 저장한다고 가정할 때 Step=1과 Step=2인 선형탐색을 이용하여 각 버켓에 저장되는 킷값의 결과를 작성하시오. 단, 버켓의 수는 5라고 가정한다.

(Step=1의 경우)

버켓	1	2	3	4	5
키값					

(Step=2의 경우)

버켓	1	2	3	4	5
키값					

12. 완전이진트리에서 높이 H와 데이터 개수 n의 관계식을 쓰시오.

13. B-A-C-D-E 순서로 키값이 삽입되었다고 가정하였을 때의 AVL트리를 그리시오.