



[취업폭격기 Zeromini 위클리 개념 폭격 #18]

📖 과목 : 자료구조론

🔥 참고문제 : 2023년 국가직 7급

😊 문제 수정 버전 : V 1.0



1. 시간 복잡도

문제: 다음 C 코드의 시간 복잡도를 설명해주세요.

```
void func(int n)
{
    int i, j, k = 0;
    for(i = 0; i < (n + 200); i++)
        for(j = 0; j < 300; j++)
            printf("k = %d\n", k++);
}
```

해설: 이 코드는 두 개의 중첩된 for 루프를 포함하고 있습니다. 첫 번째 루프는 $n+200$ 번, 두 번째 루프는 300번 실행됩니다. 따라서 전체 시간 복잡도는 $O((n+200) * 300)$ 이며, 큰 O 표기법으로는 $O(n)$ 입니다.

2.깊이 우선 탐색(DFS)

문제: 깊이 우선 탐색(DFS)의 기본 원리에 대해 설명해주세요.

해설: 깊이 우선 탐색(DFS)는 그래프나 트리를 탐색할 때, 현재 정점에서 인접한 정점 중 하나를 선택하여 계속해서 깊게 탐색하는 방법입니다. 모든 정점을 방문할 때까지 이 과정을 반복하며, 더 이상 깊게 탐색할 수 없는 경우 이전 정점으로 돌아와 다른 정점을 탐색합니다.

3.이중 연결 리스트

문제: 이중 연결 리스트의 특징에 대해 설명해주세요.

해설: 이중 연결 리스트는 각 노드가 데이터와 두 개의 링크(이전 노드와 다음 노드를 가리키는)를 가지고 있는 연결 리스트입니다. 이로 인해 양방향 탐색이 가능하며, 단순 연결 리스트에 비해 노드의 삽입 및 삭제가 효율적입니다.

4.하노이 탑 문제

문제: 하노이 탑 문제의 기본 원리와 시간 복잡도에 대해 설명해주세요.

해설: 하노이 탑 문제는 세 개의 기둥과 n 개의 원판을 사용하여, 모든 원판을 다른 기둥으로 옮기는 문제입니다. 원판은 크기 순서대로 쌓여 있으며, 한 번에 하나의 원판만 옮길 수 있습니다. 시간 복잡도는 $O(2^n)$ 입니다.

5.스택의 특징

문제: 스택의 기본적인 특징과 작동 원리에 대해 설명해주세요.

해설: 스택은 LIFO(Last In First Out) 구조를 가진 자료구조로, 가장 마지막에 들어온 데이터가 가장 먼저 나가는 특징을 가집니다. 주요 연산은 push(삽입)와 pop(제거)입니다.

6.큐의 특징

문제: 큐의 기본적인 특징과 작동 원리에 대해 설명해주세요.

해설: 큐는 FIFO(First In First Out) 구조를 가진 자료구조로, 가장 먼저 들어온 데이터가 가장 먼저 나가는 특징을 가집니다. 주요 연산은 enqueue(삽입)와 dequeue(제거)입니다.

7.배열과 연결 리스트의 차이

문제: 배열과 연결 리스트의 주요 차이점에 대해 설명해주세요.

해설: 배열은 연속적인 메모리 공간에 데이터를 저장하는 반면, 연결 리스트는 불연속적인 메모리 공간에 데이터를 저장하며, 포인터를 사용하여 연결합니다. 배열은 고정된 크기를 가지지만, 연결 리스트는 동적으로 크기가 변경됩니다.

8.이진 탐색의 원리

문제: 이진 탐색의 기본 원리에 대해 설명해주세요.

해설: 이진 탐색은 정렬된 배열에서 중간 값을 선택하여 찾고자 하는 값과 비교하는 방식입니다. 찾는 값이 중간 값보다 작으면 왼쪽 부분을, 크면 오른쪽 부분을 다시 탐색합니다.

9. 동적 프로그래밍의 특징

문제: 동적 프로그래밍의 기본 원리와 특징에 대해 설명해주세요.

해설: 동적 프로그래밍은 복잡한 문제를 작은 하위 문제로 나누어 해결하는 방법입니다. 이미 계산된 하위 문제의 결과는 저장되어 재사용되며, 이를 통해 계산 효율성을 높입니다.

10. 그래프와 트리의 차이

문제: 그래프와 트리의 주요 차이점에 대해 설명해주세요.

해설: 트리는 그래프의 한 종류로, 사이클이 없고 연결된 구조입니다. 그래프는 노드와 그 사이의 간선으로 구성되며, 사이클이 있을 수 있습니다.

11. 해시 테이블의 원리

문제: 해시 테이블의 작동 원리와 충돌 해결 방법에 대해 설명해주세요.

해설: 해시 테이블은 키를 해시 함수에 입력하여 배열의 인덱스를 얻고, 해당 인덱스에 값을 저장하는 구조입니다. 충돌이 발생할 경우, 연결 리스트나 개방 주소법 등의 방법으로 해결합니다.

12. 힙의 특징

문제: 힙의 기본적인 특징과 작동 원리에 대해 설명해주세요.

해설: 힙은 완전 이진 트리의 일종으로, 부모 노드와 자식 노드 간에 순서 속성이 유지되는 구조입니다. 주로 최대 힙과 최소 힙이 있으며, 우선순위 큐 구현에 사용됩니다.

13. 자료구조의 선택 기준

문제: 특정 애플리케이션을 개발할 때 자료구조를 선택하는 기준에 대해 설명해주세요.

해설: 자료구조를 선택할 때는 여러 가지 요소를 고려해야 합니다. 데이터의 삽입, 삭제, 탐색의 빈도, 메모리 사용량, 데이터의 크기와 종류 등의 요소를 고려하여 가장 효율적인 자료구조를 선택해야 합니다. 예를 들어, 빈번한 탐색 작업이 필요하다면 해시 테이블을, 정렬된 데이터의 빠른 검색이 필요하다면 이진 탐색 트리를 고려할 수 있습니다.

14. 재귀 함수의 특징

문제: 재귀 함수의 작동 원리와 그 특징에 대해 설명해주세요.

해설: 재귀 함수는 함수 내부에서 자신을 다시 호출하는 함수입니다. 기본적으로 탈출 조건을 포함하여, 해당 조건이 충족될 때까지 자신을 계속 호출합니다. 재귀 함수는 문제를 간결하게 표현할 수 있지만, 깊이가 너무 깊어질 경우 스택 오버플로우의 위험이 있으므로 주의가 필요합니다.

15. 그래프의 표현 방법

문제: 그래프를 표현하는 두 가지 주요 방법에 대해 설명해주세요.

해설: 그래프는 주로 인접 행렬과 인접 리스트로 표현됩니다. 인접 행렬은 2차원 배열을 사용하여 노드 간의 연결 관계를 표현하며, 공간 복잡도는 $O(V^2)$ 입니다. 인접 리스트는 각 노드에 연결된 노드의 리스트를 저장하는 방식으로, 공간 복잡도는 $O(V+E)$ 입니다. 선택 방법은 그래프의 밀집도와 사용 사례에 따라 달라집니다.

16. 밸런스 트리의 필요성

문제: 밸런스 트리와 그 필요성에 대해 설명해주세요.

해설: 밸런스 트리는 트리의 균형을 유지하는 이진 탐색 트리입니다. 균형을 유지함으로써 트리의 높이를 최소화하여 탐색, 삽입, 삭제 연산의 효율성을 보장합니다. 일반적인 이진 탐색 트리에서 데이터의 삽입 순서에 따라 트리가 한쪽으로 치우칠 수 있어, 이러한 문제를 해결하기 위해 밸런스 트리가 사용됩니다.

17. 해시 충돌의 해결 방법

문제: 해시 충돌이 발생했을 때, 그것을 해결하는 두 가지 주요 방법에 대해 설명해주세요.

해설: 해시 충돌이 발생할 경우, '개방 주소법'과 '체이닝' 방법을 사용하여 해결할 수 있습니다. 개방 주소법은 충돌이 발생한 위치에서 다음 빈 공간을 찾아 데이터를 저장하는 방식입니다. 체이닝은 각 해시 버킷에 연결 리스트를 사용하여 데이터를 연결하는 방식으로 충돌을 해결합니다.

18. B-트리의 특징

문제: B-트리의 주요 특징과 그 사용 사례에 대해 설명해주세요.

해설: B-트리는 밸런스 트리의 일종으로, 모든 리프 노드가 동일한 높이를 가지며, 각 노드의 자식 수에는 제한이 있습니다. 또한, 데이터는 오름차순으로 저장됩니다. B-트리는 디스크 기반의 저장 시스템에서 효율적인 데이터 검색을 위해 주로 사용되며, 데이터베이스 시스템과 파일 시스템에서 활용됩니다.

19. 선형 자료구조와 비선형 자료구조

문제: 선형 자료구조와 비선형 자료구조의 차이점을 설명해주세요.

해설: 선형 자료구조는 데이터 요소들이 순차적으로 배열되며, 각 요소가 정확히 하나의 전임자와 후임자를 가집니다. 예시로는 배열, 연결 리스트, 스택, 큐 등이 있습니다. 반면, 비선형 자료구조는 데이터 요소들이 계층적이거나 그래프 형태로 배열되며, 각 요소가 여러 개의 전임자나 후임자를 가질 수 있습니다. 트리와 그래프가 대표적인 예시입니다.

20. 자료구조의 중요성

문제: 왜 자료구조는 컴퓨터 과학 및 프로그래밍에서 중요한 개념인가요?

해설: 자료구조는 데이터의 효율적인 저장 및 조작을 가능하게 합니다. 올바른 자료구조를 선택하면 알고리즘의 효율성을 크게 향상시킬 수 있습니다. 또한, 메모리 사용량을 최소화하고, 실행 시간을 줄이며, 코드의 가독성과 유지 보수성을 향상시키는 데 중요한 역할을 합니다.

21. 자료구조의 분류

문제: 추상 자료형(ADT)과 구체 자료형의 차이점은 무엇인가요?

해설: 추상 자료형(ADT)은 자료구조의 수학적 모델로, 연산의 명세만을 제공하며 구현 방법은 제공하지 않습니다. 반면, 구체 자료형은 ADT의 한 가지 구체적인 구현 방법으로, 데이터의 표현 및 연산의 구현 방법을 포함합니다.

22.배열의 장단점

문제: 배열 자료구조의 주요 장점과 단점을 설명해주세요.

해설: 배열의 주요 장점은 인덱스를 통해 데이터에 빠르게 접근할 수 있다는 것입니다. 이로 인해 $O(1)$ 의 시간 복잡도로 원소에 접근할 수 있습니다. 단점은 크기가 고정되어 있어 동적으로 크기를 변경하기 어렵고, 중간에 원소를 삽입하거나 삭제할 때 다른 원소들을 이동시켜야 하는 비효율성이 있습니다.

23.연결 리스트의 사용 시나리오

문제: 연결 리스트가 배열보다 유리한 사용 사례는 무엇인가요?

해설: 연결 리스트는 동적으로 크기를 변경할 수 있기 때문에, 크기가 불규칙적으로 변하는 경우에 유리합니다. 또한, 중간에 원소를 삽입하거나 삭제하는 연산이 빈번하게 발생하는 경우, 연결 리스트는 배열보다 효율적입니다. 연결 리스트는 각 노드의 포인터만 변경하면 되므로, 원소의 이동이 필요 없습니다.

24.스택과 큐의 활용

문제: 스택과 큐 자료구조의 대표적인 활용 사례에 대해 설명해주세요.

해설: 스택은 후입선출(LIFO)의 특성을 가지므로, 함수 호출, 실행 취소, 브라우저의 뒤로 가기 기능 등에 활용됩니다. 큐는 선입선출(FIFO)의 특성을 가지므로, 데이터의 순차적 처리, 작업 스케줄링, 너비 우선 탐색 등에 활용됩니다.

25.자료구조의 분류

문제: 자료구조의 기본 분류와 각 분류의 특징을 설명하세요.

해설: 자료구조는 크게 선형 자료구조와 비선형 자료구조로 분류됩니다. 선형 자료구조에는 배열, 연결 리스트, 스택, 큐 등이 있으며, 데이터가 순차적으로 저장됩니다. 비선형 자료구조에는 트리와 그래프가 있으며, 데이터 간의 계층적 또는 네트워크 형태의 관계를 표현합니다.

26.해시 테이블

문제: 해시 테이블의 작동 원리와 충돌 해결 방법에 대해 설명하세요.

해설: 해시 테이블은 키를 값에 매핑하는 데이터 구조입니다. 키를 해시 함수에 입력하면 해시 값이 출력되며, 이 값을 인덱스로 사용하여 데이터를 저장합니다. 충돌이 발생할 경우, 연쇄법 또는 개방 주소법 등의 방법을 사용하여 충돌을 해결합니다.

27.이진 탐색 트리

문제: 이진 탐색 트리의 정의와 특징을 설명하세요.

해설: 이진 탐색 트리는 각 노드에 키가 있으며, 왼쪽 서브트리의 모든 키는 루트의 키보다 작고, 오른쪽 서브트리의 모든 키는 루트의 키보다 큰 특징을 가진 이진 트리입니다. 이러한 구조로 인해 효율적인 탐색이 가능합니다.

28.그래프의 종류

문제: 그래프의 기본 종류와 각 종류의 특징을 설명하세요.

해설: 그래프는 크게 방향 그래프와 무방향 그래프로 분류됩니다. 방향 그래프는 간선에 방

향이 있으며, 무방향 그래프는 간선에 방향이 없습니다. 또한 가중치 그래프는 간선에 가중치가 할당된 그래프를 의미합니다.

29.깊이 우선 탐색과 너비 우선 탐색

문제: 깊이 우선 탐색(DFS)과 너비 우선 탐색(BFS)의 차이점을 설명하세요.

해설: DFS는 시작 정점에서 깊게 들어가면서 탐색하는 방법이며, 스택을 사용합니다. BFS는 시작 정점에서 가까운 정점부터 탐색하는 방법이며, 큐를 사용합니다.

30.동적 프로그래밍

문제: 동적 프로그래밍의 개념과 그 특징에 대해 설명하세요.

해설: 동적 프로그래밍은 큰 문제를 작은 문제로 나누어 해결하는 방법입니다. 이미 계산된 작은 문제의 해답은 저장되어 재사용되므로 계산 효율이 높아집니다.

31.스택과 큐의 차이

문제: 스택과 큐의 기본적인 차이점과 각각의 활용 예시를 설명하세요.

해설: 스택은 LIFO(Last In First Out) 구조로, 마지막에 들어온 데이터가 먼저 나가는 구조입니다. 반면, 큐는 FIFO(First In First Out) 구조로, 먼저 들어온 데이터가 먼저 나가는 구조입니다. 스택은 괄호 검사, 후위 표기법 계산 등에 활용되며, 큐는 프린터의 인쇄 대기열, 너비 우선 탐색 등에 활용됩니다.

32.힙과 이진 탐색 트리의 차이

문제: 힙과 이진 탐색 트리의 주요 차이점을 설명하세요.

해설: 힙은 완전 이진 트리의 형태를 가지며, 부모와 자식 간의 대소 관계만을 만족하는 반면, 이진 탐색 트리는 왼쪽 자식은 부모보다 작고, 오른쪽 자식은 부모보다 큰 값을 가지는 특징을 가집니다.

33.분할 정복 전략

문제: 분할 정복 전략의 기본 원리와 그 활용 예시를 설명하세요.

해설: 분할 정복은 큰 문제를 작은 문제로 분할하여 각각을 해결한 후, 그 해답을 합쳐 원래의 문제를 해결하는 전략입니다. 병합 정렬, 퀵 정렬, 하노이의 탑 문제 등에서 활용됩니다.

34.그리디 알고리즘

문제: 그리디 알고리즘의 기본 원리와 그 특징에 대해 설명하세요.

해설: 그리디 알고리즘은 현재 상황에서 가장 좋아 보이는 선택을 하는 방법입니다. 이 방법은 항상 최적의 해답을 보장하지는 않지만, 효율적인 해답을 빠르게 찾을 수 있습니다. 활용 예시로는 거스름돈 문제, 분할 가능 배낭 문제 등이 있습니다.

35.최단 경로 알고리즘

문제: 최단 경로 알고리즘의 개념과 대표적인 알고리즘에 대해 설명하세요.

해설: 최단 경로 알고리즘은 그래프에서 두 노드 간의 최단 경로를 찾는 알고리즘입니다. 대표적인 알고리즘으로는 다익스트라 알고리즘, 벨만-포드 알고리즘 등이 있습니다.

36.백트래킹

문제: 백트래킹의 기본 원리와 그 활용 예시를 설명하세요.

해설: 백트래킹은 가능한 모든 상황을 탐색하다가, 특정 상황에서 원하는 해답을 얻을 수 없다고 판단되면 이전 상태로 돌아가는 방법입니다. 활용 예시로는 N-퀸 문제, 스도쿠 퍼즐 해결 등이 있습니다.

37.NP-완전 문제

문제: NP-완전 문제의 정의와 그 특징에 대해 설명하세요.

해설: NP-완전 문제는 모든 NP 문제가 그 문제로 변환될 수 있는 문제를 의미합니다. 현재까지는 이러한 문제들을 다항 시간 내에 해결하는 알고리즘이 알려져 있지 않습니다. 대표적인 예로는 외판원 문제, 부분 집합 합 문제 등이 있습니다.

38.동적 계획법과 분할 정복의 차이

문제: 동적 계획법과 분할 정복 전략의 주요 차이점을 설명하세요.

해설: 동적 계획법은 문제를 작은 부분 문제로 나누어 해결하며, 이미 해결된 부분 문제의 해답을 저장하여 재사용합니다. 분할 정복은 문제를 독립적인 부분 문제로 나누어 해결하며, 부분 문제의 해답을 합쳐 원래의 문제를 해결합니다.

39.그래프의 표현 방식

문제: 그래프를 표현하는 두 가지 주요 방식에 대해 설명하세요.

해설: 그래프는 주로 인접 행렬과 인접 리스트로 표현됩니다. 인접 행렬은 2차원 배열을 사용하여 노드 간의 연결 관계를 표현하며, 인접 리스트는 각 노드에 연결된 노드의 목록을 저장하는 방식입니다.

40.빅오 표기법

문제: 빅오 표기법의 정의와 그 중요성에 대해 설명하세요.

해설: 빅오 표기법은 알고리즘의 최악의 실행 시간을 표현하는 방법입니다. 이 표기법을 사용하면 알고리즘의 효율성을 객관적으로 비교할 수 있으며, 알고리즘의 성능을 예측하는 데 도움을 줍니다.