



[취업폭격기 Zeromini 위클리 개념 폭격 #2]

📖 과목 : 자료구조 - 알고리즘

🔥 참고문제 : 2023년 공무원 9급 기출문제 (알고리즘 과목)

😊 문제 수정 버전 : V 1.0



1. 알고리즘의 세 가지 기본 조건에 대해 설명하시오.

해설: 알고리즘의 기본 조건은 다음과 같습니다.

모든 명령은 모호하지 않고 명확해야 한다.

모든 명령은 실행 가능한 연산이어야 한다.

모든 명령은 유한한 시간 내에 종료되어야 한다.

2. 알고리즘의 수행 시간 분석이 중요한 이유는 무엇인가요?

해설: 알고리즘의 수행 시간 분석은 알고리즘의 효율성을 평가하는 데 중요합니다. 이를 통해 어떤 알고리즘이 더 빠르고 효율적인지 판단할 수 있습니다.

3. 너비 우선 탐색(breadth first search) 알고리즘의 동작 과정을 설명하시오.

해설: 너비 우선 탐색 알고리즘은 시작 정점을 'visited'로 표시하고 큐에 넣습니다. 그 다음, 큐에서 정점을 꺼내고, 그 정점의 인접 정점 중 아직 방문하지 않은 곳들을 'visited'로 표시하고 큐에 넣습니다. 이 과정을 큐가 비워질 때까지 반복합니다.

4. 빅오(big-oh) 표기법에 대해 설명하시오.

해설: 빅오 표기법은 알고리즘의 최악의 경우의 실행 시간을 나타내는 표기법입니다. 이는 입력 크기에 따른 알고리즘의 실행 시간 상한을 나타냅니다.

5. 선택 정렬(selection sort) 알고리즘의 동작 과정을 설명하시오.

해설: 선택 정렬 알고리즘은 배열에서 가장 작은 원소를 찾아 첫 번째 위치와 교환하고, 그 다음으로 작은 원소를 찾아 두 번째 위치와 교환하는 과정을 반복합니다. 이 과정을 통해 배열을 오름차순으로 정렬합니다.

6. 피보나치 수열을 계산하는 재귀 함수를 작성하시오.

- 해설: 피보나치 수열은 첫 번째와 두 번째 항이 1이며, 그 이후의 모든 항은 바로 앞 두 항의 합인 수열입니다. 이를 재귀 함수로 표현하면 다음과 같습니다.

```
def fibonacci(n):
    if n == 0:
        return 0
    elif n == 1:
        return 1
    else:
        return fibonacci(n-1) + fibonacci(n-2)
```

7. 점근 표기법에 대해 설명하시오.

- 해설: 점근 표기법은 알고리즘의 실행 시간이나 공간 복잡도를 표현하는 방법입니다. 이는 입력 크기가 무한대로 접근할 때 알고리즘의 성능을 비교하는 데 사용됩니다.

8. 재귀 함수를 사용하여 x의 n제곱을 계산하는 함수를 작성하시오.

- 해설: 재귀 함수를 사용하여 x의 n제곱을 계산하는 함수는 다음과 같습니다.

```
def power(x, n):
    if n == 0:
        return 1
    else:
        return x * power(x, n-1)
```

9. 힙 정렬(heap sort) 알고리즘에 대해 설명하시오.

- 해설: 힙 정렬은 완전 이진 트리인 힙을 사용하여 정렬하는 알고리즘입니다. 힙의 특성을 이용하여 배열에서 가장 큰(또는 가장 작은) 원소를 효율적으로 찾아 정렬하는 데 사용됩니다.

10. 퀵 정렬(quick sort) 알고리즘의 동작 과정을 설명하시오.

- 해설: 퀵 정렬은 분할 정복 알고리즘의 한 종류로, 피벗을 기준으로 배열을 두 부분으로 분할한 후 각 부분을 재귀적으로 정렬하는 알고리즘입니다.

11. 재귀 알고리즘의 장단점에 대해 설명하시오.

- 해설: 재귀 알고리즘은 문제를 간단하게 표현할 수 있고, 코드가 직관적이고 이해하기 쉬운 것이 장점입니다. 하지만 재귀적 호출로 인해 메모리 사용량이 증가하고, 함수 호출에 따른 오버헤드가 발생하는 것이 단점입니다.

12. 동적 계획(dynamic programming) 알고리즘에 대해 설명하시오

- 해설: 동적 계획법은 복잡한 문제를 간단한 여러 개의 문제로 나누어 푸는 방법입니다. 이 방법은 각 문제의 해결책을 저장하고 이를 재사용함으로써 계산 시간을 줄입니다.

13. 이진 검색(binary search) 알고리즘의 동작 과정을 설명하시오.

- 해설: 이진 검색은 정렬된 배열에서 특정 원소를 찾는 알고리즘입니다. 배열의 중간 원소를 선택하고, 찾으려는 값과 비교하여 찾으려는 값이 더 작으면 왼쪽 하위 배열에서, 더 크면 오른쪽 하위 배열에서 검색을 계속합니다.

14. 그래프의 깊이 우선 탐색(depth-first search) 알고리즘에 대해 설명하시오

- 해설: 깊이 우선 탐색은 그래프의 한 정점에서 시작하여 인접한 정점을 재귀적으로 탐색하는 알고리즘입니다. 이 방법은 스택을 사용하여 아직 탐색하지 않은 정점을 저장합니다.

15. 해시 테이블(hash table)에 대해 설명하시오.

- 해설: 해시 테이블은 키를 값에 매핑하는 데이터 구조입니다. 해시 함수를 사용하여 키를 해시 테이블의 인덱스로 변환하고, 이 인덱스를 사용하여 값을 저장하거나 검색합니다.

16. 다익스트라(Dijkstra) 알고리즘에 대해 설명하시오.

- 해설: 다익스트라 알고리즘은 그래프에서 한 정점에서 다른 모든 정점까지의 최단 경로를 찾는 알고리즘입니다. 이 알고리즘은 우선순위 큐를 사용하여 다음으로 탐색할 정점을 선택합니다.

17. 머지 정렬(merge sort) 알고리즘의 동작 과정을 설명하시오.

- 해설: 머지 정렬은 배열을 두 부분으로 나눈 후 각 부분을 재귀적으로 정렬하고, 정렬된 두 부분을 합치는 알고리즘입니다.

18. 트리(tree) 데이터 구조에 대해 설명하시오.

- 해설: 트리는 노드와 이들을 연결하는 에지로 구성된 비선형 데이터 구조입니다. 트리는 하나의 루트 노드를 가지며, 루트 노드 아래에는 서브트리가 있습니다.

19. 그리디 알고리즘(greedy algorithm)에 대해 설명하시오.

- 해설: 그리디 알고리즘은 각 단계에서 최선의 선택을 하는 알고리즘입니다. 이 방법은 항상 최적의 결과를 보장하지는 않지만, 계산 시간을 크게 줄일 수 있습니다.

20. 레드-블랙 트리(red-black tree)에 대해 설명하시오.

- 해설: 레드-블랙 트리는 각 노드에 색깔을 부여하여 트리의 균형을 유지하는 이진 탐색 트리입니다. 이 트리는 검색, 삽입, 삭제 연산을 효율적으로 수행할 수 있습니다.