

※ 객관식 문제(40문제 중 20문제, 각 2점)

1. 트리(Tree)의 기본 정의로 옳은 것은?

- ① 순환을 허용하는 무방향 그래프
- ② 루트에서 리프 노드까지의 경로 길이가 모두 같은 그래프
- ③ 한 개의 루트 노드와 0개 이상의 자식 노드들로 구성된 계층적 자료 구조
- ④ 서로 다른 노드들 간의 모든 경로가 존재하는 그래프

2. 이진 탐색 트리(Binary Search Tree)의 특징으로 옳지 않은 것은?

- ① 왼쪽 서브트리의 모든 노드 값은 루트 노드 값보다 작다.
- ② 오른쪽 서브트리의 모든 노드 값은 루트 노드 값보다 크다.
- ③ 중복된 값이 허용된다.
- ④ 중위 순회(Inorder Traversal)를 하면 오름차순으로 정렬된 값을 얻는다.

3. 다음 중 트리 순회 방법이 아닌 것은?

- ① 전위 순회(Preorder Traversal)
- ② 중위 순회(Inorder Traversal)
- ③ 후위 순회(Postorder Traversal)
- ④ 순환 순회(Circular Traversal)

4. 완전 이진 트리(Complete Binary Tree)의 정의로 옳은 것은?

- ① 모든 내부 노드가 두 개의 자식 노드를 가진다.
- ② 모든 리프 노드가 같은 깊이에 있다.
- ③ 마지막 레벨을 제외한 모든 레벨이 완전히 채워져 있다.
- ④ 모든 노드가 왼쪽에 자식 노드를 가진다.

5. 높이가 3인 완전 이진 트리의 최대 노드 수는?

- ① 3
- ② 7
- ③ 15
- ④ 31

6. AVL 트리의 주요 특징은 무엇인가?

- ① 각 노드의 왼쪽과 오른쪽 서브트리의 높이 차이가 1 이하로 유지된다.
- ② 모든 노드가 최대 두 개의 자식을 가진다.
- ③ 중위 순회를 통해 정렬된 배열을 얻는다.
- ④ 최대 높이가 $\log(n)$ 이하로 제한된다.

7. 다음 중 최대 힙(Max Heap)의 특징으로 옳은 것은?

- ① 루트 노드의 값이 항상 가장 작다.
- ② 모든 자식 노드의 값이 루트 노드의 값보다 크다.

- ③ 완전 이진 트리의 형태를 유지한다.
- ④ 모든 리프 노드의 값이 부모 노드의 값보다 크다.

8. 선택 정렬(Selection Sort)에 대한 설명으로 옳은 것은?

- ① 데이터의 분할과 병합을 통해 정렬을 수행한다.
- ② 매 단계에서 가장 작은(또는 큰) 요소를 선택하여 앞으로 이동시킨다.
- ③ 항상 $O(n \log n)$ 의 시간 복잡도를 가진다.
- ④ 평균적으로 $O(n)$ 의 시간 복잡도를 가진다.

9. 다음은 선택 정렬의 일반적인 과정 중 하나입니다. 올바른 과정은 무엇인가요?

- ① 각 단계에서 현재 위치의 요소와 그 다음 요소를 비교하여 정렬한다.
- ② 각 단계에서 현재 위치의 요소를 임의의 위치의 요소와 교환한다.
- ③ 각 단계에서 가장 작은(또는 큰) 요소를 찾아 현재 위치의 요소와 교환한다.
- ④ 각 단계에서 현재 위치의 요소를 왼쪽 서브트리와 오른쪽 서브트리로 나눈다.

10. 선택 정렬의 특징으로 옳지 않은 것은?

- ① 제자리(in-place) 정렬 알고리즘이다.
- ② 안정 정렬(stable sort)이다.
- ③ 비교 기반 정렬 알고리즘이다.
- ④ 입력 배열의 순서와 상관없이 항상 동일한 시간 복잡도를 가진다.

11. 선택 정렬의 장점으로 올바른 것은?

- ① 모든 경우에서 $O(n \log n)$ 의 시간 복잡도를 가진다.
- ② 적은 메모리 공간을 필요로 한다.
- ③ 데이터가 거의 정렬된 경우 매우 효율적이다.
- ④ 데이터의 크기에 관계없이 항상 $O(n)$ 의 시간 복잡도를 가진다.

12. 배열 [64, 25, 12, 22, 11]을 선택 정렬로 정렬할 때, 첫 번째 단계 후 배열의 상태는 무엇인가요?

- ① [11, 25, 12, 22, 64]
- ② [64, 25, 12, 22, 11]
- ③ [11, 64, 25, 22, 12]
- ④ [64, 22, 12, 25, 11]

13. 선택 정렬은 왜 안정 정렬이 아닌가요?

- ① 추가적인 메모리 공간이 필요하다.
- ② 동일한 요소의 상대적 순서를 유지하지 않는다.
- ③ 비교 기반 정렬 알고리즘이 아니다.
- ④ 데이터가 거의 정렬된 경우에 비효율적이다.

14. 삽입 정렬(Insertion Sort)의 기본 개념은 무엇입니까?

- ① 데이터의 분할과 병합을 통해 정렬한다.
- ② 매 단계에서 가장 작은 요소를 선택하여 앞으로 이동시킨다.
- ③ 정렬된 부분과 정렬되지 않은 부분으로 나누어, 정렬되지 않은 부분의 첫 요소를 정렬된 부분의 적절한 위치에 삽입한다.
- ④ 각 단계에서 인접한 두 요소를 비교하여 교환한다.

15. 다음 중 삽입 정렬의 일반적인 과정은 무엇입니까?

- ① 현재 위치의 요소를 그 다음 요소와 교환한다.
- ② 현재 위치의 요소를 가장 작은 요소와 교환한다.
- ③ 현재 위치의 요소를 정렬된 부분의 적절한 위치에 삽입한다.
- ④ 현재 위치의 요소를 왼쪽 서브트리와 오른쪽 서브트리로 나눈다.

16. 배열 [4, 3, 2, 10, 12, 1, 5, 6]을 삽입 정렬로 정렬할 때, 첫 번째 단계 후 배열의 상태는 무엇입니까?

- ① [3, 4, 2, 10, 12, 1, 5, 6]
- ② [4, 3, 2, 10, 12, 1, 5, 6]
- ③ [3, 2, 4, 10, 12, 1, 5, 6]
- ④ [2, 3, 4, 10, 12, 1, 5, 6]

17. 삽입 정렬의 특징 중 옳지 않은 것은 무엇입니까?

- ① 제자리(in-place) 정렬 알고리즘입니다.
- ② 비교 기반 정렬 알고리즘입니다.
- ③ 데이터가 거의 정렬된 경우 매우 효율적입니다.
- ④ 입력 데이터의 순서에 상관없이 항상 $O(n \log n)$ 의 시간 복잡도를 가집니다.

18. 삽입 정렬의 장점 중 옳은 것은 무엇입니까?

- ① 대규모 데이터 정렬에 적합합니다.
- ② 간단하고 구현이 용이합니다.
- ③ 항상 $O(n \log n)$ 의 시간 복잡도를 가집니다.
- ④ 분산 시스템에서 효율적입니다.

19. 배열 [5, 4, 3, 2, 1]을 삽입 정렬로 정렬할 때, 세 번째 단계 후 배열의 상태는 무엇입니까?

- ① [4, 3, 5, 2, 1]
- ② [3, 4, 5, 2, 1]
- ③ [2, 3, 4, 5, 1]
- ④ [1, 2, 3, 4, 5]

20. 합병 정렬(Merge Sort)의 기본 개념은 무엇입니까?

- ① 배열을 반으로 나눈 뒤 각 부분을 정렬하여 병합한다.
- ② 가장 작은(또는 큰) 요소를 선택하여 앞으로 이동시킨다.

- ③ 각 단계에서 인접한 두 요소를 비교하여 교환한다.
- ④ 배열을 여러 그룹으로 분할하여 정렬한 뒤 합친다.

21. 합병 정렬의 일반적인 과정은 무엇입니까?

- ① 현재 위치의 요소를 그 다음 요소와 교환한다.
- ② 현재 위치의 요소를 가장 작은 요소와 교환한다.
- ③ 배열을 반으로 나눈 뒤 각 부분을 정렬하여 병합한다.
- ④ 현재 위치의 요소를 정렬된 부분의 적절한 위치에 삽입한다.

22. 배열 [4, 3, 2, 10, 12, 1, 5, 6]을 합병 정렬로 정렬할 때, 첫 번째 단계 후 배열의 상태는 무엇입니까?

- ① [3, 4, 2, 10, 12, 1, 5, 6]
- ② [4, 3, 2, 10, 12, 1, 5, 6]
- ③ [3, 2, 4, 10, 12, 1, 5, 6]
- ④ [2, 3, 4, 10, 12, 1, 5, 6]

23. 합병 정렬의 특징 중 옳지 않은 것은 무엇입니까?

- ① 제자리(in-place) 정렬 알고리즘입니다.
- ② 비교 기반 정렬 알고리즘입니다.
- ③ 입력 데이터의 순서에 상관없이 항상 $O(n \log n)$ 의 시간 복잡도를 가집니다.
- ④ 분할 정복(divide and conquer) 방법을 사용합니다.

24. 합병 정렬의 장점 중 옳은 것은 무엇입니까?

- ① 대규모 데이터 정렬에 적합합니다.
- ② 간단하고 구현이 용이합니다.
- ③ 항상 $O(n \log n)$ 의 시간 복잡도를 가집니다.
- ④ 분산 시스템에서 효율적입니다.

25. 퀵 정렬(Quick Sort)의 기본 개념은 무엇입니까?

- ① 배열을 반으로 나눈 뒤 각 부분을 정렬하여 병합한다.
- ② 기준 요소(pivot)를 선택하고, 기준보다 작은 요소는 왼쪽으로, 큰 요소는 오른쪽으로 분할하여 정렬한다.
- ③ 가장 작은(또는 큰) 요소를 선택하여 앞쪽으로 이동시킨다.
- ④ 각 단계에서 인접한 두 요소를 비교하여 교환한다.

26. 퀵 정렬의 일반적인 과정은 무엇입니까?

- ① 현재 위치의 요소를 그 다음 요소와 교환한다.
- ② 현재 위치의 요소를 가장 작은 요소와 교환한다.
- ③ 기준 요소(pivot)를 선택하고, 기준보다 작은 요소는 왼쪽으로, 큰 요소는 오른쪽으로 분할하여 정렬한다.
- ④ 현재 위치의 요소를 정렬된 부분의 적절한 위치에 삽입한다.

27. 배열 [4, 3, 2, 10, 12, 1, 5, 6]을 퀵 정렬로 정렬할 때, 첫 번째 단계 후 배열의 상태는 무엇입니까?

- ① [3, 4, 2, 10, 12, 1, 5, 6]
- ② [4, 3, 2, 10, 12, 1, 5, 6]
- ③ [3, 2, 4, 10, 12, 1, 5, 6]
- ④ [2, 3, 4, 10, 12, 1, 5, 6]

28. 퀵 정렬의 특징 중 옳지 않은 것은 무엇입니까?

- ① 제자리(in-place) 정렬 알고리즘입니다.
- ② 비교 기반 정렬 알고리즘입니다.
- ③ 입력 데이터의 순서에 상관없이 항상 $O(n \log n)$ 의 시간 복잡도를 가집니다.
- ④ 분할 정복(divide and conquer) 방법을 사용합니다.

29. 퀵 정렬의 장점 중 옳은 것은 무엇입니까?

- ① 대규모 데이터 정렬에 적합합니다.
- ② 간단하고 구현이 용이합니다.
- ③ 항상 $O(n \log n)$ 의 시간 복잡도를 가집니다.
- ④ 분산 시스템에서 효율적입니다.

30. 힙 정렬(Heap Sort)의 기본 개념은 무엇입니까?

- ① 배열을 반으로 나눈 뒤 각 부분을 정렬하여 병합한다.
- ② 가장 작은(또는 큰) 요소를 선택하여 앞으로 이동시킨다.
- ③ 최대 힙 또는 최소 힙을 구성한 뒤, 힙의 루트를 추출하여 정렬한다.
- ④ 각 단계에서 인접한 두 요소를 비교하여 교환한다.

31. 힙 정렬의 일반적인 과정은 무엇입니까?

- ① 현재 위치의 요소를 그 다음 요소와 교환한다.
- ② 현재 위치의 요소를 가장 작은 요소와 교환한다.
- ③ 최대 힙 또는 최소 힙을 구성한 뒤, 힙의 루트를 추출하여 정렬한다.
- ④ 현재 위치의 요소를 정렬된 부분의 적절한 위치에 삽입한다.

32. 배열 [4, 3, 2, 10, 12, 1, 5, 6]을 힙 정렬로 정렬할 때, 첫 번째 단계 후 배열의 상태는 무엇입니까?

- ① [3, 4, 2, 10, 12, 1, 5, 6]
- ② [4, 3, 2, 10, 12, 1, 5, 6]
- ③ [3, 2, 4, 10, 12, 1, 5, 6]
- ④ [2, 3, 4, 10, 12, 1, 5, 6]

33. 힙 정렬의 특징 중 옳지 않은 것은 무엇입니까?

- ① 제자리(in-place) 정렬 알고리즘입니다.
- ② 비교 기반 정렬 알고리즘입니다.
- ③ 입력 데이터의 순서에 상관없이 항상 $O(n \log n)$ 의 시간 복잡도를 가집니다.
- ④ 힙 자료 구조를 사용하여 정렬을 수행합니다.

34. 힙 정렬의 장점 중 옳은 것은 무엇입니까?

- ① 대규모 데이터 정렬에 적합합니다.
- ② 간단하고 구현이 용이합니다.
- ③ 항상 $O(n \log n)$ 의 시간 복잡도를 가집니다.
- ④ 최악의 경우에도 $O(n \log n)$ 의 시간 복잡도를 가집니다.

35. 기수 정렬(Radix Sort)의 기본 개념은 무엇입니까?

- ① 배열을 반으로 나눈 뒤 각 부분을 정렬하여 병합한다.
- ② 가장 작은(또는 큰) 요소를 선택하여 앞으로 이동시킨다.
- ③ 각 자릿수(혹은 특정한 자릿수)를 기준으로 정렬하는 비교 정렬 알고리즘입니다.
- ④ 각 단계에서 인접한 두 요소를 비교하여 교환한다.

36. 기수 정렬의 일반적인 과정은 무엇입니까?

- ① 각 자릿수(혹은 특정한 자릿수)를 기준으로 정렬하여 임시 배열에 저장한 뒤, 원래 배열에 복사합니다.
- ② 현재 위치의 요소를 가장 작은 요소와 교환한다.
- ③ 현재 위치의 요소를 그 다음 요소와 교환한다.
- ④ 현재 위치의 요소를 정렬된 부분의 적절한 위치에 삽입한다.

37. 배열 [4, 3, 2, 10, 12, 1, 5, 6]을 기수 정렬로 정렬할 때, 첫 번째 단계 후 배열의 상태는 무엇입니까?

- ① [3, 4, 2, 10, 12, 1, 5, 6]
- ② [4, 3, 2, 10, 12, 1, 5, 6]
- ③ [3, 2, 4, 10, 12, 1, 5, 6]
- ④ [2, 3, 4, 10, 12, 1, 5, 6]

38. 기수 정렬의 특징 중 옳지 않은 것은 무엇입니까?

- ① 제자리(in-place) 정렬 알고리즘입니다.
- ② 비교 기반 정렬 알고리즘입니다.
- ③ 입력 데이터의 순서에 상관없이 항상 $O(nk)$ 의 시간 복잡도를 가집니다.
- ④ 자릿수가 많은 경우 메모리를 많이 차지할 수 있습니다.

39. 기수 정렬의 장점 중 옳은 것은 무엇입니까?

- ① 대규모 데이터 정렬에 적합합니다.
- ② 간단하고 구현이 용이합니다.
- ③ 비교 연산을 사용하지 않기 때문에 일반적인 비교 정렬보다 빠릅니다.
- ④ 추가적인 메모리 공간이 필요하지 않습니다.

40. 기수 정렬은 어떤 상황에서 다른 정렬 알고리즘보다 우수합니까?

- ① 데이터의 범위가 작은 경우
- ② 데이터의 범위가 큰 경우
- ③ 데이터가 이미 정렬되어 있는 경우
- ④ 데이터가 무작위로 섞인 경우

※ 주관식 문제

1. 아래는 이진 트리를 배열 자료형으로 표현한 것이다. 질문에 답하세요. (10점)

0	1	2	3	4	5	6	7	8	9	10	11	12
	A	B	C	D	E	F	G	H	I	J	K	

- (1) “D”의 부모 노드를 구하는 수식을 활용하여 부모노드를 구하세요. (수식 반드시 표기)
- (2) “E”의 왼쪽, 오른쪽 자식 노드를 구하는 수식을 활용하여 왼쪽, 오른쪽 자식 노드를 구하세요. (수식 반드시 표기)

2. 아래는 배열 자료형에 입력된 데이터를 합병 정렬로 정렬하는 과정을 그림과 함께 설명하세요. (15점)

13	36	54	23	53	42	12	44
----	----	----	----	----	----	----	----

3. 아래의 데이터를 이용하여 기수 정렬의 LSD 방식으로 정렬하는 과정을 그림과 함께 설명하세요. (15점)

- 데이터: [153, 262, 37, 598, 433, 855]

4. Big-O 표기법에 대해서 아는 대로 설명하세요. (20점)

