

알고리즘

1. 알고리즘의 조건에 대한 설명으로 옳은 것만을 모두 고르면?

- ㄱ. 모든 명령은 모호하지 않고 명확해야 한다.
ㄴ. 모든 명령은 실행 가능한 연산이어야 한다.
ㄷ. 모든 명령은 반복적으로 무한히 실행되어야 한다.

- ① ㄱ
② ㄱ, ㄴ
③ ㄴ, ㄷ
④ ㄱ, ㄴ, ㄷ

2. 알고리즘의 수행 시간 분석에 대한 설명으로 옳지 않은 것은?

- ① 알고리즘의 수행 시간은 컴퓨터 성능에 관계없이 명확하게 정의되어야 한다.
② 알고리즘의 시간복잡도는 수행하는 기본적인 연산 횟수를 입력 크기에 대한 함수로 표현한다.
③ 최선의 경우의 알고리즘 수행 시간은 모든 입력의 수행 시간에 대한 상한이 된다.
④ 알고리즘의 수행 시간 분석에는 최악의 경우, 평균의 경우, 최선의 경우가 있다.

3. 다음은 그래프에서 너비 우선 탐색(breadth first search) 알고리즘이 동작하는 과정이다. (가) ~ (다)에 들어갈 내용을 바르게 연결한 것은?

[단계 1] 시작 정점을 'visited'로 표시하고 (가) 에(서) (나) 한다.

[단계 2] (가) 에(서) 정점을 (다) 하고, 제거한 정점의 인접 정점 중 아직 방문하지 않은 곳들은 'visited'로 표시하고 (가) 에(서) (나) 한다.

[단계 3] (가) 가/이 비워질 때까지 [단계 2]를 반복한다.

- | (가) | (나) | (다) |
|---------|---------|---------|
| ① Queue | Dequeue | Enqueue |
| ② Queue | Enqueue | Dequeue |
| ③ Stack | Pop | Push |
| ④ Stack | Push | Pop |

4. 입력 크기 n 에 대한 수행 횟수를 빅오(big-oh) 표기법으로 표현했을 때 옳지 않은 것은?

- ① $n^4 + 2^n + 5 \rightarrow O(n^4)$
② $3n^2 + 8n + 7 \rightarrow O(n^2)$
③ $2n^2 + 2n \log n + 4n \rightarrow O(n^2)$
④ $3n + 5n \log n + 2 \rightarrow O(n \log n)$

5. 다음 의사코드(pseudo code)가 설명하는 정렬 알고리즘은?

○ 입력: 크기가 n 인 배열 A
○ 출력: 정렬된 배열 A

```
for i = 0 to n-2 {  
    min = i  
    for j = i+1 to n-1 {  
        if(A[j] < A[min])  
            min = j  
    }  
    A[i] ↔ A[min]  
}  
return A
```

- ① 버블 정렬(bubble sort)
② 삽입 정렬(insertion sort)
③ 선택 정렬(selection sort)
④ 퀵 정렬(quick sort)

6. 다음 fib() 함수는 피보나치 수열을 계산한다. fib(6)을 실행할 때, fib() 함수의 호출 횟수는? (단, fib(6)의 호출은 제외한다)

```
int fib(int n)  
{  
    if(n == 0) return 0;  
    if(n == 1) return 1;  
    return ( fib(n-1) + fib(n-2) );  
}
```

- ① 21번
② 22번
③ 23번
④ 24번

7. (가) ~ (다)에 들어갈 점근 표기법은?

- $n \geq n_0$ 인 모든 n 에 대해 $c_1g(n) \leq f(n) \leq c_2g(n)$ 을 만족하는 양의 상수 c_1, c_2, n_0 가 존재하기만 하면 $f(n) = \boxed{\text{(가)}}$ 이다.
- $n \geq n_0$ 인 모든 n 에 대해 $f(n) \leq cg(n)$ 인 조건을 만족하는 양의 상수 c 와 n_0 가 존재하기만 하면 $f(n) = \boxed{\text{(나)}}$ 이다.
- $n \geq n_0$ 인 모든 n 에 대해 $f(n) \geq cg(n)$ 을 만족하는 양의 상수 c 와 n_0 가 존재하기만 하면 $f(n) = \boxed{\text{(다)}}$ 이다.

- | | (가) | (나) | (다) |
|---|----------------|----------------|----------------|
| ① | $\Omega(g(n))$ | $O(g(n))$ | $\Theta(g(n))$ |
| ② | $\Omega(g(n))$ | $\Theta(g(n))$ | $O(g(n))$ |
| ③ | $\Theta(g(n))$ | $O(g(n))$ | $\Omega(g(n))$ |
| ④ | $\Theta(g(n))$ | $\Omega(g(n))$ | $O(g(n))$ |

8. 다음 재귀함수 power()는 x^n 을 계산한다. 빈칸에 들어갈 내용은?

```
double power(double x, int n)
{
    if (n == 0) return 1;
    return x *  
}
```

- ① power(x, n);
- ② power(x, n-1);
- ③ power(x-1, n);
- ④ power(x-1, n-1);

9. 힙 정렬(heap sort)을 수행하기 위해 다음 데이터를 왼쪽부터 차례대로 하나씩 삽입하여 최소힙(min heap)을 구성하였다. 이후 루트를 한 번 삭제하고 최소힙 특성을 유지하기 위해 재조정 한 후, 루트의 왼쪽 자식 노드의 값은?

4, 6, 8, 3, 7, 1, 5, 2, 9

- ① 3
- ② 4
- ③ 5
- ④ 6

10. 다음 조건으로 퀵 정렬(quick sort)을 수행할 때, 처음 데이터 교환이 발생하는 배열의 인덱스 쌍은?

- 데이터를 오름차순으로 정렬한다.
- low는 왼쪽에서 오른쪽으로 탐색할 때, high는 오른쪽에서 왼쪽으로 탐색할 때 사용되는 변수이다.
- 정렬할 데이터(A[])는 다음과 같으며, 피벗(pivot)의 초기값은 A[0]이고, low와 high의 초기값은 각각 1과 8이다.

i	0	1	2	3	4	5	6	7	8
A[i]	23	5	91	10	21	35	18	42	78

- ① 1, 8
- ② 2, 6
- ③ 2, 7
- ④ 3, 8

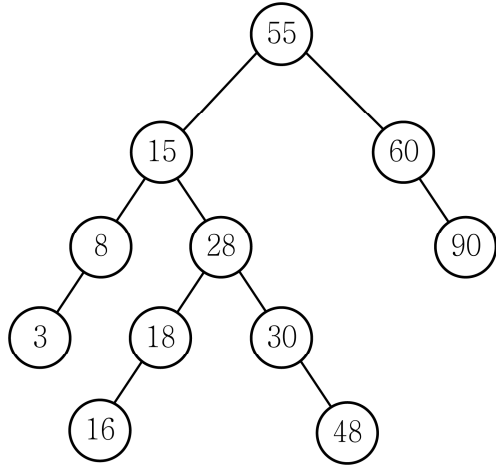
11. 재귀 알고리즘에 대한 설명으로 옳지 않은 것은?

- ① 재귀호출은 함수가 자기 자신을 호출하는 것이다.
- ② 재귀함수는 재귀호출이 끝나는 종료 조건이 있어야 한다.
- ③ 재귀함수로 작성된 병합 정렬(merge sort) 알고리즘은 반복문을 이용하여 구현할 수 있다.
- ④ 재귀함수는 실행 시간과 메모리 공간 사용 측면에서 반복문보다 효율성이 높다.

12. 동적 계획(dynamic programming) 알고리즘에 대한 설명으로 옳지 않은 것은?

- ① 최적화 문제에 적용할 수 있는 알고리즘이다.
- ② 상위 문제의 해를 분할하여 하위 문제의 해를 구한다.
- ③ 최적해를 구하는 방법을 재귀적으로 정의한다.
- ④ 한 번 계산된 부분 문제들의 해는 재사용을 위해 저장된다.

13. 다음 이진 탐색 트리(binary search tree)에서 키 48을 검색할 때, 방문하는 노드의 순서는?



- ① 55 - 15 - 28 - 30 - 48
 ② 55 - 60 - 90 - 15 - 28 - 30 - 48
 ③ 55 - 15 - 8 - 3 - 28 - 18 - 16 - 30 - 48
 ④ 55 - 15 - 60 - 8 - 28 - 90 - 3 - 18 - 30 - 16 - 48

14. 다음 정렬 알고리즘의 평균 시간복잡도를 바르게 연결한 것은?

(가) 선택 정렬	(나) 삽입 정렬
(다) 퀵 정렬	(라) 버블 정렬

- | | (가) | (나) | (다) | (라) |
|---|----------|---------------|---------------|---------------|
| ① | $O(n)$ | $O(n^2)$ | $O(n \log n)$ | $O(n^2)$ |
| ② | $O(n)$ | $O(n^2)$ | $O(n^2)$ | $O(n \log n)$ |
| ③ | $O(n^2)$ | $O(n \log n)$ | $O(n)$ | $O(n^2)$ |
| ④ | $O(n^2)$ | $O(n^2)$ | $O(n \log n)$ | $O(n^2)$ |

15. (가)에 들어갈 내용은?

- (가) 는 사용되는 문자가 N개일 때, 문자열 검색을 빠르게 실행할 수 있도록 설계된 N진 트리이다.
 ○ (가) 를 이용한 문자열 검색은 루트 노드에서 시작하여 탐색키의 첫 번째 문자에 연관된 링크를 따라 노드를 찾아가고 그 노드에서 다시 두 번째 문자에 연관된 링크를 따라 노드를 찾아가는 과정을 검색이 완료될 때까지 반복한다.

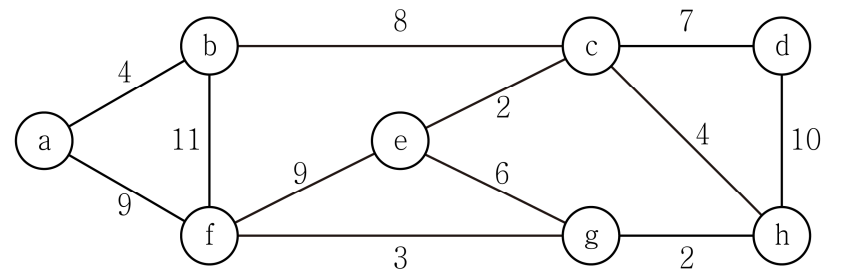
- ① AVL 트리(AVL tree)
 ② B+ 트리(B+ tree)
 ③ 레드-블랙 트리(red-black tree)
 ④ 트라이(trie)

16. 다음에서 설명하는 알고리즘 설계 기법에 해당하지 않는 것은?

문제를 해결할 때 여러 경우 중 하나를 결정해야 할 때마다 현재 순간에 최적이라고 생각되는 것을 선택하면서 문제의 최종해에 도달한다.

- ① 다익스트라(Dijkstra) 알고리즘
 ② 프림(Prim) 알고리즘
 ③ 플로이드-워셜(Floyd-Warshall) 알고리즘
 ④ 허프만(Huffman) 코딩 알고리즘

17. 다음 그래프에서 Kruskal 알고리즘을 사용하여 최소 신장 트리(minimum spanning tree)를 찾을 때, 최소 신장 트리의 간선을 나열한 것은?



- ① (a,b) (a,f) (b,c) (c,d) (c,e) (f,g) (g,h)
 ② (a,b) (a,f) (b,c) (c,d) (e,g) (f,g) (g,h)
 ③ (a,b) (b,c) (c,d) (c,e) (c,h) (f,g) (g,h)
 ④ (a,b) (b,c) (c,d) (c,e) (e,g) (f,g) (g,h)

18. 알고리즘 설계 기법 중 분할정복(divide-and-conquer)에 대한 설명으로 옳지 않은 것은?

- ① 이진 탐색(binary search)을 위해 분할정복을 적용할 수 없다.
 ② 분할정복을 사용한 대표적인 정렬 방법에는 병합 정렬(merge sort)이 있다.
 ③ 문제를 작은 문제로 분할하고 그 문제들의 해를 병합한다.
 ④ 분할된 문제들이 서로 중첩되지 않는 경우에 적합하다.

19. 입력으로 길이 n 의 텍스트 문자열(T)과 길이 m 의 패턴 문자열(P)이 있을 때, 문자열 매칭(string matching) 알고리즘에 대한 설명으로 옳지 않은 것은?

- ① 브루트-포스(brute-force) 방식의 수행 시간은 $O(mn)$ 이다.
- ② 라빈-카프(Rabin-Karp) 알고리즘은 P의 해시(hash) 값을 이용한다.
- ③ 보이어-무어(Boyer-Moore) 알고리즘은 P의 각 문자를 왼쪽에서 오른쪽으로 스캔하면서 T와 비교한다.
- ④ KMP(Knuth-Morris-Pratt) 알고리즘은 P의 각 문자에 대해 매칭 실패 시 비교를 다시 시작할 위치를 계산해 놓는다.

20. 백트래킹(backtracking)에 대한 설명으로 옳은 것만을 모두 고르면?

- ㄱ. 너비 우선 탐색(breadth first search) 방식을 기반으로 한다.
- ㄴ. 문제를 해결하는 과정에서 해를 더 이상 얻지 못하는 상황이 되면 직전 상황으로 되돌아가서 다시 해를 탐색하는 기법이다.
- ㄷ. 적용할 수 있는 대표적인 문제로는 미로 찾기, N-여왕 문제 등이 있다.

- ① ㄱ
- ② ㄴ
- ③ ㄴ, ㄷ
- ④ ㄱ, ㄴ, ㄷ