

만든야: 조 현 준

- 성균관대학교 정보공학 전공
- CISA, CISSP, 정보보안기사
- 前, 데카르트고시학원 전산직 전임강사
- 現, 지안공무원학원 전산직 전임강사
- 現, (주)지안에듀 정보보안(산업)기사 전임강사

- TopSpot 자료구조론 이론편/기술편
- TopSpot 알기사 정보보안기사(산업)기사 필기
- TopSpot 알기사 정보보안기사(산업)기사 실기
- TopSpot 정보보호론
- TopSpot 정보보호론 기술문제집

유튜브 기술해설 강의 목록(링크) ; 조현준 정보보호론 검색
https://www.youtube.com/playlist?list=PLaR0eJQDBqV8Mb3h4hdxwrQ_nRtY0aCQWB
 유튜브 기술해설 강의 목록(링크) ; 조현준 자료구조론 검색
https://www.youtube.com/playlist?list=PLaR0eJQDBqV_-MpiNRHK0aPVlnrMeUMSE

2023년 국가직 9급 알고리즘

2023년 4월 8일 시행

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

정답과 해설

알고리즘은 어떤 경우에도 한정된 수의 단계 뒤에는 반드시 종료해야 한다.

 ②

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.

정답과 해설

알고리즘의 수행시간 분석에는 최악(O), 평균(Θ), 최선(Ω)의 경우가 있다. 최선의 경우의 알고리즘 수행 시간은 모든 입력의 수행 시간에 대한 하한이 된다.

 ③

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 |

정답과 해설

너비우선탐색은 각 정점을 방문할 때마다 그 정점은 큐에 저장된다. 한 인접 리스트가 끝나면 큐에서 한 정점을 꺼내 그 정점의 인접 리스트에 있는 정점들을 같은 방법으로 계속 조사한다. Enqueue는 큐에 삽입하는 것이고, Dequeue는 큐에서 삭제하는 것이다.



②

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 $3n + 5n \log n + 2 \rightarrow O(n \log n)$

정답과 해설

빅오는 점근적 상한을 의미한다. $n^4 + 2^n + 5 \rightarrow O(2^n)$



①

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)

정답과 해설

선택정렬기법은 레코드 이동 횟수를 줄이려는 목적에서 개발되었으며, 자료 중에서 최소값 또는 최대값을 선택해 가면서 리스트의 처음이나 마지막으로 이동하는 방식이다. 보기는 선택 정렬 의사코드이다.



③

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

정답과 해설

종료조건($n == 0$, $n == 1$)을 체크하면서 피보나치 수열을 그리서 풀면 쉽다. fib(6) 호출을 제외하고 총 24번 호출한다.



④

7. ○△×

(가) ~ (다)에 들어갈 점근 표기법은?

- $n \geq n_0$ 인 모든 n 에 대해 $c_1 g(n) \leq f(n) \leq c_2 g(n)$ 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 $f(n) \leq c g(n)$ 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 $f(n) \geq c g(n)$ 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 $\Theta(g(n))$ | $\Omega(g(n))$ | $O(g(n))$ |

정답과 해설

알고리즘의 수행시간 분석에 사용하는 평균(Θ), 최악(O), 최선(Ω)에 대한 정의이다.



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);

정답과 해설

재귀함수로 x^n 을 계산하기 위해서는 첨자 n 은 점차 줄어(n-1), 종료 조건으로 수렴하도록 구성하면 된다.



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 재조정된 후, 루트의 왼쪽 자식 노드의 값은?

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

- ① 3
- ② 4
- ③ 5
- ④ 6

정답과 해설

차례대로 하나씩 삽입하여 최소힙을 구성하여 순서대로 나열하면 다음과 같다. (1, 2, 3, 4, 7, 8, 5, 6, 9)

루트(1)를 삭제하고, 끝에 있는 원소(9)를 루트로 이동 후에 조정하면 다음과 같다. (2, 4, 3, 6, 7, 8, 5, 9)

루트의 왼쪽 자식 노드는 4가 된다.



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

정답과 해설

피벗 23을 기준으로 오름차순 정렬이므로, low는 A[1]~A[8] 방향으로 23보다 큰 원소 찾는다. 반면에 high는 A[8]~A[1] 방향으로 23보다 작은 원소를 찾는다. 이후 찾은 원소 A[2]와 A[6]을 교환한다.



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.

정답과 해설

순환(recursion)이란 어떤 알고리즘이나 함수가 자기 자신을 호출하여 문제를 해결하는 프로그래밍 기법이다. 순환함수는 단계를 진행함에 따라 종료 조건에 접근해 가야하며, 입력 값이 완료 조건에 만족할 때 자신을 호출하지 않고 출력 값을 계산한다. 순환적 프로그램을 사용하면 알고리즘의 시간 복잡도와 공간 복잡도는 보통 더 증가한다.

④

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.

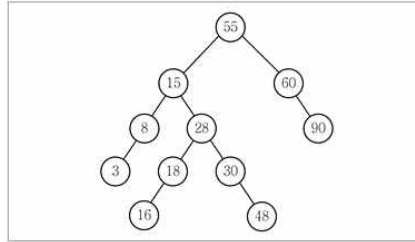
정답과 해설

동적 계획(dynamic programming) 기법은 Bottom Up 접근 방식으로, 작은 단계의 문제를 해결하고, 그 답을 저장한 후 계속 그것을 이용하여 더 큰 문제들을 해결해 나간다. 즉 하위 문제의 해를 분할하여 상위 문제의 해를 구한다.

②

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

정답과 해설

이진 탐색 트리는 공백이 가능한 이진트리로, 공백이 아니라면 특정 부모노드를 기준으로 자노드 값이 크면 오른쪽에, 작으면 왼쪽에 배치할 수 있는 트리이다. 48을 검색할 때도 키값을 비교하여 55, 15, 28, 30, 48 순서로 찾아간다.

①

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 $O(n^2)$ | $O(n^2)$ | $O(n \log n)$ | $O(n^2)$ |

정답과 해설

정렬 종류	최선	평균	최악
선택정렬	$O(n^2)$	$O(n^2)$	$O(n^2)$
거품정렬	$O(n)$	$O(n^2)$	$O(n^2)$
삽입정렬	$O(n^2)$	$O(n^2)$	$O(n^2)$
퀵정렬	$O(n \log_2 n)$	$O(n \log_2 n)$	$O(n^2)$

④

15. ○△×

(가)에 들어갈 내용은?

- (가) 는 사용되는 문자가 N개일 때, 문자열 검색을 빠르게 실행할 수 있도록 설계된 N진 트리아이다.
- (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 트리아(trie)

정답과 해설

트리아(trie)는 키 탐색을 위해 키 값을 직접 표현하지 않고 키를 구성하는 문자나 숫자의 순서로 키 값을 표현한 자료구조로, m-진 탐색트리이나 m-원 균형트리는 아니다.



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

정답과 해설

* Greedy Method(탐욕법, 욕심쟁이 방법)

- ① 각 선택 과정에서 현 단계에서 최선이라고 할 수 있는 선택을 하면서, 결과적으로 전체적인 최적해를 구할 수 있을 것이라는 희망적인 전략 방법이다.
- ② 희망적인 전략이란 각 단계마다 선택한 최적해가 전체 최적해를 만들지 못할 수도 있음을 내포한다. 즉, 탐욕법(희망전략)으로 해를 구할 수 없는 문제도 있다.
- ③ 탐욕법에는 Kruskal, Prim 알고리즘 등이 있다. 위상 정렬 및 Huffman트리에도 greedy method를 이용하는 등 탐욕

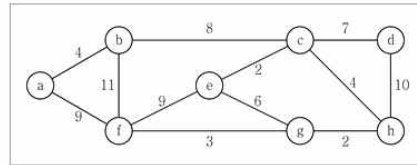
법은 광범위한 프로그래밍 문제에도 적용될 수 있다.

- ④ 모든 정점 간의 최단 경로 비용을 구하기 위한 플로이드-워셜(Floyd-Warshall) 알고리즘은 탐욕법에 해당되지 않는다.



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)

정답과 해설

Kruskal 알고리즘은 오름차순 기법을 이용한다. 순서대로 나열하면 다음과 같다.

(c,e)	(g,h)	(f,g)	(a,b)	(c,h)	(c,g)	(c,d)	(b,c)
2	2	3	4	4	거부	7	8

(c,g)는 사이클을 형성하여 거부된다.



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.

정답과 해설

문제 해결을 위해 대상을 여러 개의 작은 부분으로 나누어 각 문제를 해결한 후 그 결과를 합병하여 원하는 해를 구하는 방법이다. 즉, 분할과 정복 기법(Divide and Conquer)을 순환적으로 적용한다. 이진 검색은 분할정복을 적용한다.



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 P 의 해시(hash) 값을 이용한다.
- ③ 보이어-무어(Boyer-Moore) 알고리즘은 P 의 각 문자를 왼쪽에서 오른쪽으로 스캔하면서 T 와 비교한다.
- ④ KMP(Knuth-Morris-Pratt) 알고리즘은 P 의 각 문자에 대해 매칭 실패 시 비교를 다시 시작할 위치를 계산해 놓는다.

정답과 해설

- ① 문자열 매칭 알고리즘은 전체 문자열 $string[n]$ 에서 부분 문자열 $pattern[m]$ 이 존재하는지, 존재한다면 $pattern$ 이 $string$ 에서 시작되는 인덱스 값을 리턴하는 알고리즘이다.
- ② 라빈-카프(Rabin-Karp) 알고리즘은 전체 문자열 $string$ 내에서 부분 문자열 $pattern$ 을 찾아내는데, Hash Algorithm을 이용하여 $string$ 과 $pattern$ 의 해시값을 비교하는 방법이다.
- ③ KMP(Knuth-Morris-Pratt) 알고리즘은 Knuth, Morris, Pratt이 함께 개발한 String Matching Algorithm이다. 메인 문자열에 부분 문자열이 존재하는지를 파악하고, 존재한다면 시작되는 위치를 알린다. Matching 실패 시에는 건너뛰어도 되는 문자 개수를 계산한다.
- ④ 보이어-무어(Boyer-Moore) 알고리즘은 오른쪽에서 왼쪽으로 스캔하면서 비교한다.



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

정답과 해설

- ① 뒤추적(backtracking) 기법은 어떤 노드에서 해답이 나올 가능성을 점검한 후, 해답이 나올 가능성이 없다고 판단되면 그 노드의 이전 노드로 되돌아가서("backtracking") 다른 노드로부터 해답을 다시 찾으려는 기법이다.
- ② 너비 우선 탐색은 큐를 이용하고, 깊이 우선 탐색은 스택을 이용하는 백트래킹 방법이다.
- ③ N -여왕 문제는 크기가 $N \times N$ 인 체스판 위에 퀸 N 개가 서로를 공격할 수 없게 놓는 경우의 수를 구하는 문제이다.

