

알고리즘

1. 정점이 6개인 무방향 완전 그래프의 간선의 개수는?

- ① 6
② 12
③ 15
④ 21

2. 큐에 다음 함수를 적용한 결과로 (가) ~ (다)에 들어갈 내용을 바르게 연결한 것은?

enqueue(e): 큐의 제 위치에 객체 e를 삽입한다.
dequeue(): 큐의 제 위치에서 객체를 삭제한다. 빈 큐일 경우에는 “error”를 출력한다.

연산순서	함수	출력	큐 저장공간
1	dequeue()	“error”	()
2	enqueue(1)		(1)
3	enqueue(2)		(1, 2)
4	enqueue(5)		(1, 2, 5)
5	dequeue()		(가)
6	enqueue(7)		(나)
7	dequeue()		(다)

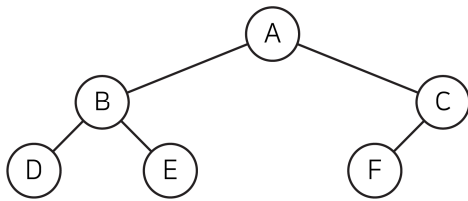
(가)

(나)

(다)

- ① (1, 2) (1, 2, 7) (1, 2)
② (2, 5) (2, 5, 7) (5, 7)
③ (2, 5) (5, 7) (7)
④ (2, 5, 1) (2, 5, 1, 7) (5, 1, 7, 2)

3. 다음 최대 힙에 대한 설명으로 옳지 않은 것은?



- ① 완전 이진 트리이어야 한다.
② 노드 A는 가장 큰 키 값을 가져야 한다.
③ 노드 B는 노드 F보다 큰 키 값을 가져야 한다.
④ 노드 B와 노드 D가 같은 키 값을 가질 수 있다.

4. 그리디 알고리즘으로 최적의 해가 보장되지 않는 것은?

- ① 허프만(Huffman) 코딩
② 최소 신장 트리
③ 다익스트라(Dijkstra) 최단 경로
④ 0-1 배낭(knapsack) 문제

5. 다음 C 코드에서 사용한 재귀 알고리즘의 시간 복잡도는? (단, n은 배열의 크기이다)

```
int S(int A[ ], int n){  
    if( n <= 0 ) return 0;  
    else return A[n-1] + S(A, n/2);  
}
```

- ① $\Theta(\log n)$
② $\Theta(n)$
③ $\Theta(n \log n)$
④ $\Theta(n^2)$

6. 1부터 n까지 합을 구하는 다음 파이썬 프로그램의 빈칸에 들어갈 내용은? (단, n은 $n \geq 1$ 인 정수이다)

```
def sum(n):  
    ret = 0  
    for  :  
        ret += i  
    return ret
```

- ① `i in range(-1, n+1)`
② `i in range(0, n)`
③ `i in range(1, n)`
④ `i in range(1, n+1)`

7. 연산 이론상의 문제 유형에 대한 설명으로 옳지 않은 것은?

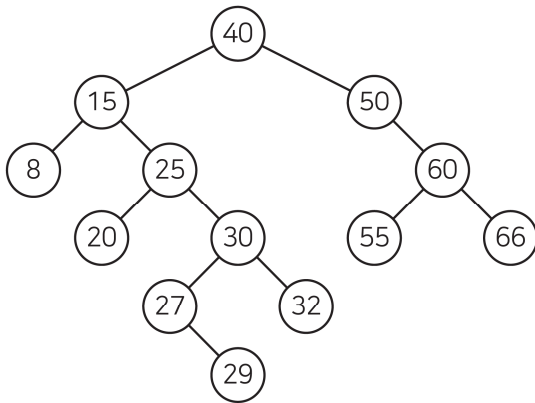
- ① P는 다항식 시간에 해결할 수 있는 문제군이다.
- ② NP는 비결정론적 다항식 시간에 해결할 수 있는 문제군이다.
- ③ 어떤 NP-Complete 문제에 대해서 다항식 시간에 해결할 수 있는 알고리즘을 찾아내면, 다른 모든 NP-Complete 문제도 다항식 시간에 해를 구할 수 있다.
- ④ NP-Complete 문제들 중에는 NP에 속하지 않는 문제도 있다.

8. 다음 fib() 함수는 피보나치 수열을 계산하는 C 코드이다. fib(17)을 실행할 때 fib(8)과 fib(10)이 각각 호출되는 횟수를 바르게 연결한 것은?

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

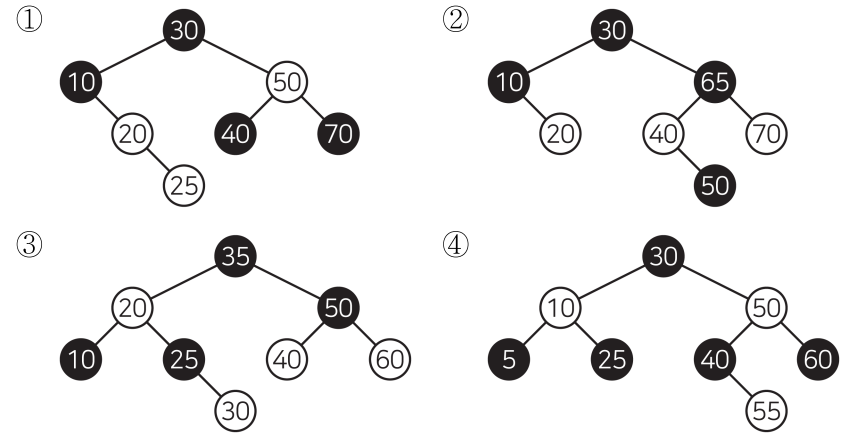
- | | <u>fib(8)</u> | <u>fib(10)</u> |
|---|---------------|----------------|
| ① | 52번 | 13번 |
| ② | 52번 | 21번 |
| ③ | 55번 | 13번 |
| ④ | 55번 | 21번 |

9. 다음 이진 검색 트리에서 노드 이동 또는 포인터 변경의 횟수를 최소화하면서 노드를 삭제하는 과정의 설명으로 옳지 않은 것은?



- ① 노드 8을 삭제하려면 노드 15에서 노드 8을 가리키는 포인터를 NIL(또는 NULL)로 변경한다.
- ② 노드 25를 삭제하려면 그 자리에 노드 30을 놓는다.
- ③ 노드 50을 삭제하려면 노드 40이 노드 60을 직접 가리키도록 포인터를 변경한다.
- ④ 노드 60을 삭제하려면 그 자리에 노드 55 또는 노드 66을 놓는다.

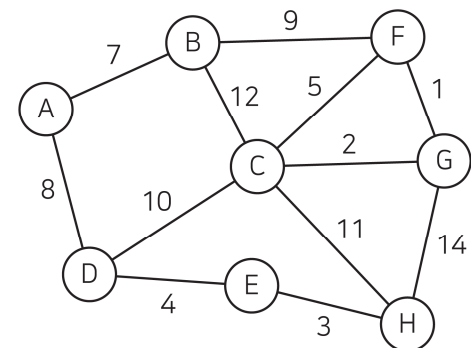
10. 레드-블랙 트리로 옳은 것은? (단, 외부 노드(leaf)는 표시하지 않고, 블랙 노드는 검정색으로, 레드 노드는 흰색으로 나타낸 것이다)



11. 해시 테이블에 대한 설명으로 옳지 않은 것은?

- ① 충돌이 발생하지 않는다면 상수 시간 $O(1)$ 에 검색과 저장 가능하다.
- ② 키 값을 입력으로 받아 해시 테이블상의 주소를 리턴하는 해시 함수는 나누기 방법뿐만 아니라 곱하기 방법으로도 만들 수 있다.
- ③ 저장 원소의 총수를 해시 테이블의 크기로 나눈 적재율(load factor)은 해시 테이블의 성능에 중요한 영향을 미친다.
- ④ 충돌을 해결하기 위한 별도 체이닝과 오픈 주소 방법은 모두 추가적인 저장 공간을 필요로 하지 않는다.

12. 프림(Prim) 알고리즘을 적용하여 다음 그래프의 정점 A에서 시작하는 최소 신장 트리를 구할 때, 마지막으로 선택되는 정점은?



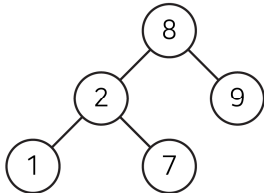
- | | |
|-----|-----|
| ① C | ② F |
| ③ G | ④ H |

19. 그림과 같은 상태의 크기 13인 해시 테이블에 이중(double) 해싱 방법으로 원소 14를 삽입하고 이어서 원소 20을 삽입할 때, 원소 20의 저장 주소는? (단, 해시 함수 $h(k,i) = (h_1(k) + i \cdot h_2(k)) \bmod 13$, 충돌 횟수 $i = 0, 1, 2, \dots$ 이고, $h_1(k) = k \bmod 13$, $h_2(k) = 1 + (k \bmod 11)$ 이다)

주소	
0	
1	66
2	
3	
4	69
5	18
6	
7	72
8	
9	
10	
11	50
12	

- ① 6 ② 8
③ 9 ④ 10

20. 다음 AVL 트리에 대해 노드 4, 5, 6을 차례대로 삽입한 결과는? (단, 리밸런싱을 위한 회전 횟수는 최소로 한다)



- ① ②
- ③ ④

21. 다음 파이썬 프로그램의 실행 결과는?

```

a=[29,88,60,26,40,74,43,22,61,40]
def partition():
    i = -1
    n = len(a)
    for j in range(0, n-2):
        if a[j] <= a[n-1]:
            i += 1
            b = a[i]
            a[i] = a[j]
            a[j] = b
    b = a[i+1]
    a[i+1] = a[n-1]
    a[n-1] = b

partition()
print(a)

```

- ① 29 22 26 40 40 74 43 88 61 60
② 29 22 40 26 40 74 43 88 61 60
③ 29 26 22 40 40 74 43 60 61 88
④ 29 26 40 22 40 74 43 88 61 60

22. 점근적 표기의 정의로 옳지 않은 것은?

- ① $O(g(n)) = \{f(n) \mid \text{모든 } n \geq n_0 \text{에 대하여 } f(n) \leq c \cdot g(n) \text{인 양의 상수 } c \text{와 } n_0 \text{가 존재한다}\}$
② $\Omega(g(n)) = \{f(n) \mid \text{모든 } n \geq n_0 \text{에 대하여 } c \cdot g(n) \leq f(n) \text{인 양의 상수 } c \text{와 } n_0 \text{가 존재한다}\}$
③ $\Theta(g(n)) = \{f(n) \mid \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0\}$
④ $\omega(g(n)) = \{f(n) \mid \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = \infty\}$

23. 배열 A를 오름차순으로 정렬하기 위해 5, 3, 1의 간격을 사용하여 셸(Shell) 정렬을 수행할 때, 간격 3에 대한 처리까지 수행을 완료한 후의 배열 A의 상태로 옳은 것은?

i	0	1	2	3	4	5	6	7	8	9	10
A[i]	9	18	26	21	4	17	13	2	6	24	8

- ① 4 9 13 26 6 8 18 24 2 17 21
- ② 6 4 2 8 13 9 18 17 21 24 26
- ③ 9 2 6 13 4 17 21 8 26 24 18
- ④ 9 18 26 4 17 21 2 6 13 8 24

24. 다음 C 코드에 해당하는 문자열 탐색 방법은? (단, s1은 입력 문자열, s2는 s1에서 찾고자 하는 패턴 문자열이고, n1은 s1의 길이, n2는 s2의 길이를 나타낸다)

```
int match(char *s1, char *s2) {
    int pt = 0, p = 0;
    while(pt != n1 && p != n2) {
        if (s1[pt] == s2[p]) {
            pt++;
            p++;
        } else {
            pt = pt - p + 1;
            p = 0;
        }
    }
    if (p == n2) return pt-p;
    else return -1;
}
```

- ① 브루트-포스(brute-force) 알고리즘
- ② 라빈-카프(Rabin-Karp) 알고리즘
- ③ 오토마타를 이용한 매칭 알고리즘
- ④ 보이어-무어(Boyer-Moore) 알고리즘

25. 다음은 텍스트에서 패턴과 일치하는 문자열을 찾기 위한 KMP(Knuth Morris Pratt) 알고리즘에서의 전처리 과정을 구현한 파이썬 프로그램이다. 패턴이 abcbabca인 경우, 아래 표의 (가) ~ (라)에 들어갈 숫자를 바르게 연결한 것은?

```
def preprocess(pn: str):
    pt = 1
    k = 0
    pi = [0] * (len(pn) + 1)
    pi[pt-1] = -1
    pi[pt] = 0
    while pt != len(pn):
        if pn[pt] == pn[k]:
            k += 1
            pt += 1
            pi[pt] = k
        elif k == 0:
            pt += 1
            pi[pt] = k
        else:
            k = pi[k]
    print(pi)
pattern = 'abcbabca'
preprocess(pattern)
```

i	0	1	2	3	4	5	6	7	8
pn[i]	a	b	c	b	a	b	c	a	
pi[i]	-1	0	0	0	0	(가)	(나)	(다)	(라)

- (가) (나) (다) (라)
- ① 1 1 1 1
- ② 1 2 3 1
- ③ 1 2 3 0
- ④ 3 3 3 0