알고리즘

1. 다음은 1 이상인 x에 대해 1부터 x까지의 합을 계산하는 C 함수이다. (가)에 들어갈 코드는?

```
int sum(int x)
{
    if (x <= 1) return 1;
    return x + (가);
}
```

① x + 1
② x - 1
③ sum(x + 1)
④ sum(x - 1)

2. 다음 설명에 해당하는 알고리즘은?

○ 문자열 매칭 알고리즘이다.
○ 주어진 패턴과 텍스트에서 사용된 알파벳을 이용해 불일치 문자(bad character) 이동표를 만든다.
○ 패턴을 이용해서 일치 접미부(good suffix) 이동표를 만든다.

① 다익스트라(Dijkstra) 알고리즘
② 라빈－카프(Rabin-Karp) 알고리즘
③ 보이어-무어(Boyer-Moore) 알고리즘
④ 플로이드－워셜(Floyd-Warshall) 알고리즘

3. 분할 정복(divide and conquer) 방식의 정렬 알고리즘만을 모두 고르면?

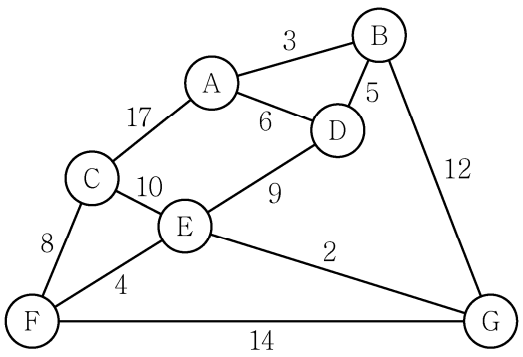
ㄱ. 버블 정렬(bubble sort)
ㄴ. 병합 정렬(merge sort)
ㄷ. 퀵 정렬(quick sort)

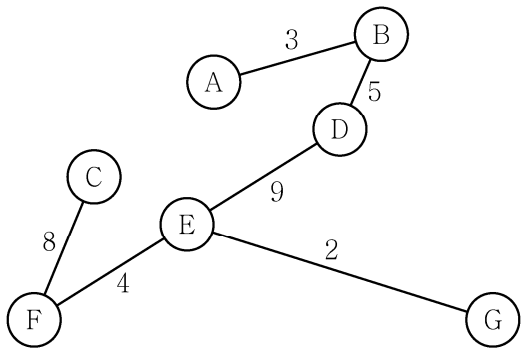
① ㄴ
② ㄱ, ㄴ
③ ㄱ, ㄷ
④ ㄴ, ㄷ

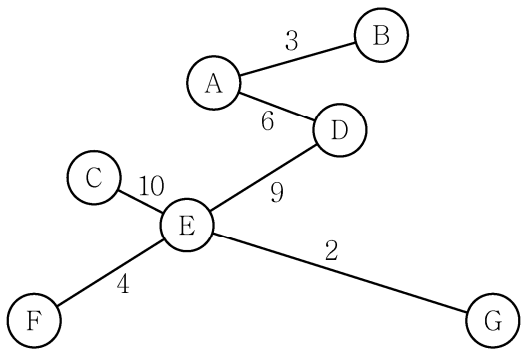
4. 빅오(O) 표기법에 대한 설명으로 옳지 않은 것은?

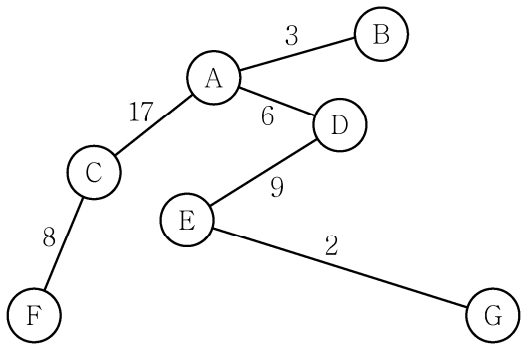
① $O(n\log n)$은 2를 포함한다.
② $O(n\log n)$은 $3n\log n + 2n + 1$을 포함한다.
③ $O(n\log n)$은 $3n + \log n + 2$를 포함하지 않는다.
④ $O(n\log n)$은 $n^2 + 2n\log n + 3$을 포함하지 않는다.

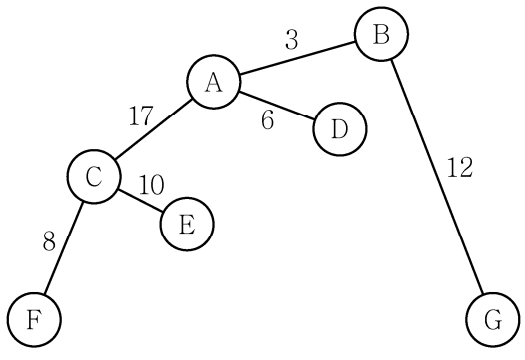
5. 다음 그래프에서 크루스칼(Kruskal) 알고리즘을 사용하여 만든 최소 비용 신장 트리(minimum cost spanning tree)는?



①


②


③


④


6. 충분히 큰 n에 대해서 수행 시간이 가장 많이 걸리는 시간 복잡도는?

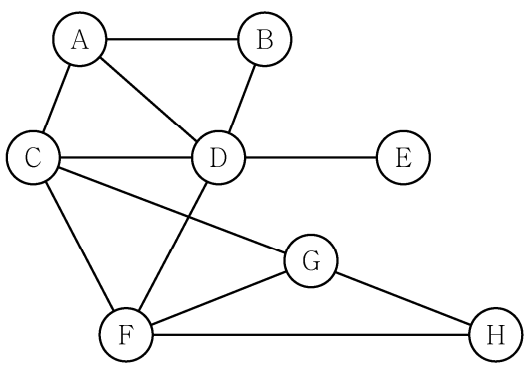
① $\Theta(2^n)$
② $\Theta(n^3)$
③ $\Theta(n!)$
④ $\Theta(n\log n)$

7. 다음 의사코드에 해당하는 알고리즘 설계기법은?

```
f[ ]: 모든 값이 0으로 초기화된 정수 배열
fib(n)
{
    if (f[n] == 0) {
        if (n == 1 or n == 2) f[n] = 1;
        else f[n] = fib(n - 1) + fib(n - 2);
    }
    return f[n];
}
```

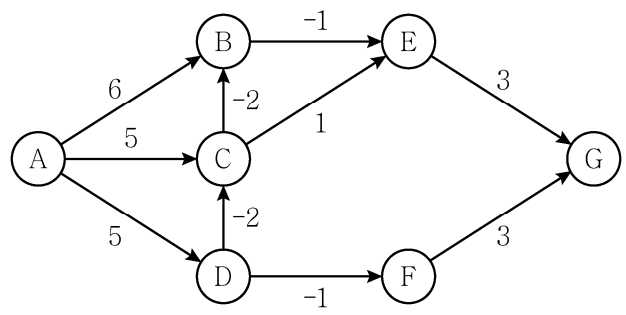
① 그리디(greedy) 알고리즘
② 유전자(genetic) 알고리즘
③ 분기 한정(branch and bound) 기법
④ 동적 프로그래밍(dynamic programming)

8. 다음 그래프의 A 정점부터 너비 우선 탐색(BFS, breadth first search)을 할 때, 가능한 정점의 방문 순서가 아닌 것은?



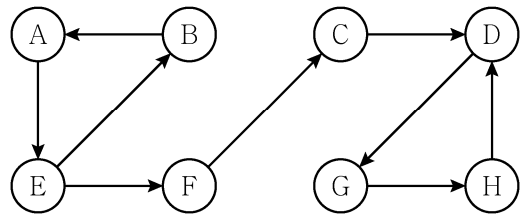
① A→B→C→D→F→G→E→H
② A→B→D→C→E→F→G→H
③ A→C→B→D→F→G→E→H
④ A→C→D→E→B→F→H→G

9. 다음 방향 그래프에 벨만-포드(Bellman-Ford) 알고리즘을 적용한 후, 각 정점과의 최단 거리 값을 바르게 연결한 것은? (단, 시작 정점은 A 정점이다)



	A	B	C	D	E	F	G
①	0	1	1	5	−1	−1	3
②	0	1	3	5	0	4	3
③	0	3	2	2	3	2	6
④	0	3	5	5	6	4	7

10. 다음 방향 그래프에서 2개 이상의 정점이 포함되어 있는 강한 연결 요소(strongly connected component)의 개수는?



① 1
② 2
③ 3
④ 4

11. 다음 배열에서 버블 정렬 알고리즘을 사용하여 오름차순 정렬했을 때, 자리바꿈의 총횟수는?

배열	65	40	80	15

오름차순 방향 →

① 3
② 4
③ 5
④ 6

12. 다음 표와 같이 분말이 있을 때, 40 kg 무게까지 허용가능한 배낭에 최대 이익을 얻을 수 있도록 분말을 넣는 알고리즘의 C 프로그램이 아래와 같다. (가), (나)에 들어갈 코드와 출력값을 바르게 연결한 것은? (단, 배낭에 각 분말 일부만 넣을 수도 있다)

분말 종류	보유량(kg)	이익
A	10	60
B	18	90
C	25	100
D	15	120

```
#include <stdio.h>

int main() {
    double wgt[] = {10, 18, 25, 15};
    double val[] = {60, 90, 100, 120};
    double ratio[4] = {}, W = 40, max_r,
           totalVal = 0.0;
    int i, max_i;

    for (i = 0; i < 4; i++)
        ratio[i] = (가);
    while (W > 0) {
        max_r = -1.0;
        max_i = -1;

        for (i = 0; i < 4; i++) {
            if (wgt[i] > 0 && ratio[i] > max_r) {
                max_r = ratio[i];
                max_i = i;
            }
        }
        if (max_i == -1) break;
        if (W >= wgt[max_i]) {
            W -= wgt[max_i];
            totalVal += val[max_i];
        }
        else {
            totalVal += val[max_i] * (나);
            break;
        }
        wgt[max_i] = 0;
    }
    printf("%.1f\n", totalVal);
    return 0;
}
```

	(가)	(나)	출력값
①	val[i] / wgt[i]	W / wgt[max_i]	255.0
②	val[i] / wgt[i]	wgt[max_i] / W	255.0
③	val[i] / wgt[i]	W / wgt[max_i]	220.0
④	wgt[i] / val[i]	W / wgt[max_i]	220.0

13. 다음 C 프로그램의 실행 결과에 포함되지 않는 것은?

```
#include <stdio.h>

int count = 0;

void f(int n, char from, char tmp, char to){
    count++;
    if (n == 1)
        printf("%d: 원판 %d를(을) %c에서 %c로 이동\n
", count, n, from, to);
    else {
        f(n - 1, from, to, tmp);
        printf("%d: 원판 %d를(을) %c에서 %c로 이동\n
", count, n, from, to);
        f(n - 1, tmp, from, to);
    }
}

int main() {
    f(3, 'a', 'b', 'c');
    return 0;
}
```

① 4: 원판 1를(을) c에서 b로 이동
② 4: 원판 3를(을) a에서 c로 이동
③ 5: 원판 1를(을) b에서 a로 이동
④ 7: 원판 1를(을) a에서 c로 이동

14. 비어있는 이진 탐색 트리(binary search tree)에 다음 키값이 순서대로 입력되면, 15를 찾기 위해 방문해야 하는 노드의 개수는?

50, 40, 30, 35, 20, 15, 25, 10

① 3
② 4
③ 5
④ 6

15. 다음 배열에서 보간 탐색(interpolation search)으로 58을 찾고자 할 때, 첫 번째 탐색 위치는? (단, 배열에서 위치의 차이는 값의 차이에 비례한다는 가정하에 탐색 위치를 계산하며 소수점 이하는 반올림한다)

위치	0	1	2	3	4	5	6	7	8	9
배열	3	7	12	22	32	58	67	80	87	89

① 2
② 4
③ 6
④ 8

16. 다음과 같이 전체 버킷 개수가 13개이고 버킷당 1개의 슬롯을 가지는 비어있는 해시 테이블에 값 <7, 20, 28, 46, 81, 67, 4>를 순서대로 해시 함수를 사용하여 저장하였을 때, 버킷 번호 4에 저장되는 값은? (단, 해시 함수로 $h(x) = x \bmod 13$을 사용하며, 충돌 해결은 개방 주소 방법의 선형 조사법(linear probing)을 적용한다)

버킷 번호	슬롯
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	

① 4
② 46
③ 67
④ 81

17. 다음 두 문자열 A와 B의 편집 거리(edit distance)는? (단, 허용하는 문자열 연산은 삽입, 삭제, 교체 연산이다)

○ 문자열 A: algorithm
○ 문자열 B: anthem

① 5
② 6
③ 7
④ 9

18. 다음 배열에 대해 아래 알고리즘을 적용하여 정렬하고자 한다. 3번째 for 루프를 수행할 때, 배열 내에서 교환되는 두 값은?

위치	0	1	2	3	4	5	6	7	8	9
배열	9	21	54	32	77	45	19	83	12	3

```
Sort(A[], n): // n은 입력배열 A의 크기이다.
    for last ← (n - 1) downto 1
        A[0 ... last] 중 가장 큰 수 A[k]를 찾는다.
        A[k]와 A[last]의 값을 교환한다.
```

① 3, 54
② 21, 45
③ 21, 54
④ 32, 77

19. 다음과 같이 오름차순 정렬을 수행하는 알고리즘은?

초기 상태	5	20	17	6	2	13	10
1단계	5	20	17	6	2	13	10
2단계	5	17	20	6	2	13	10
3단계	5	6	17	20	2	13	10
4단계	2	5	6	17	20	13	10
5단계	2	5	6	13	17	20	10
6단계	2	5	6	10	13	17	20

① 병합 정렬
② 삽입 정렬
③ 선택 정렬
④ 퀵 정렬

20. 문자에 대한 빈도수가 다음과 같을 때, exam을 허프만(Huffman) 코드로 작성하면 비트 수는?

문자	a	b	c	e	m	x
빈도수	8	5	3	10	6	1

① 10
② 11
③ 12
④ 14