

1. 알고리즘의 시간복잡도 분석에 영향을 줄 수 있는 요인으로 옳은 것만을 <보기>에서 모두 고른 것은?

<보기>
 ㄱ. 프로그래밍 언어
 ㄴ. 주요 연산의 시행 횟수
 ㄷ. 입력의 크기
 ㄹ. 프로그래머의 역량

- ① ㄱ, ㄴ
 ② ㄴ, ㄷ
 ③ ㄱ, ㄷ, ㄹ
 ④ ㄴ, ㄷ, ㄹ

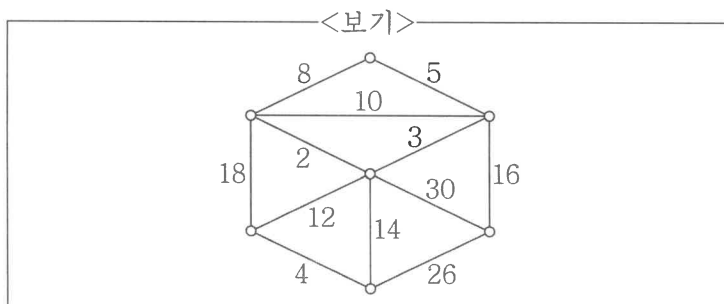
2. 시간복잡도에 대한 점근적 표기법 중 가장 옳지 않은 것은?

- ① $f(n) = \Theta(g(n))$ 이면 $f(n) = O(g(n))$ 이다.
 ② $f(n) = \Theta(g(n))$ 이면 $f(n) = \Omega(g(n))$ 이다.
 ③ $f(n) = O(g(n))$ 이면 $f(n) \neq \Omega(g(n))$ 이다.
 ④ $f(n) = O(g(n))$ 이고 $g(n) = O(h(n))$ 이면 $f(n) = O(h(n))$ 이다.

3. 분할 정복(divide-and-conquer) 기법에 대한 설명으로 가장 옳은 것은?

- ① 문제를 더 이상 나눌 수 없을 때까지 작은 문제로 분할한 후, 그 결과를 결합하여 해결한다.
 ② 문제를 해결할 때 항상 현재 단계에서 최적의 선택을 반복한다.
 ③ 이미 계산된 결과를 저장하여 동일한 계산을 반복하지 않도록 한다.
 ④ 모든 가능한 경우를 탐색하여 최적해를 찾는 방식이다.

4. 무방향 가중 그래프가 <보기>와 같이 주어져 있다. <보기>에서 간선 위의 수는 그 간선의 가중치를 의미한다. 이 그래프의 최소 신장 트리(MST, minimum spanning tree)를 구했을 때, 포함된 간선의 가중치의 합은?



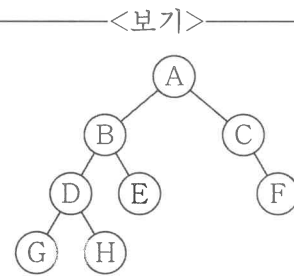
- ① 32
 ② 34
 ③ 38
 ④ 42

5. <보기>의 숫자 배열에 버블 정렬(bubble sort) 알고리즘을 사용하여 오름차순으로 정렬하려고 한다. 첫 번째 패스 이후 배열의 상태로 가장 옳은 것은?

<보기>
 7 1 3 5 2 4 9 6 8

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

6. <보기>의 트리를 후위 순회(postorder traversal)한 경우 노드의 방문 순서로 가장 옳은 것은?



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

7. <보기>의 의사코드(pseudo code)가 설명하는 정렬 알고리즘으로 가장 옳은 것은? (단, 입력은 크기가 n인 배열 A이고, 출력은 정렬된 배열 A이다.)

<보기>

```

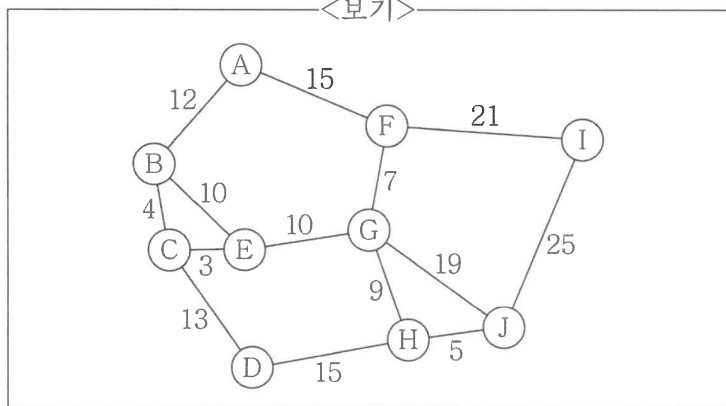
for i = 2 to n {
    key = A[i]
    j = i - 1

    while j > 0 and A[j] > key
        A[j + 1] = A[j]
        j = j - 1

    A[j + 1] = key
}
```

- ① 삽입 정렬(insertion sort)
 ② 선택 정렬(selection sort)
 ③ 병합 정렬(merge sort)
 ④ 퀵 정렬(quick sort)

8. 무방향 가중 그래프가 <보기>와 같이 주어져 있다. 그래프에서 간선 위의 수는 그 간선의 가중치를 의미한다. 이 그래프의 정점 A에서 다른 모든 정점까지의 최단 경로의 길이(가중치의 합)로 가장 옳은 것은?



- ① 201 ② 206
③ 211 ④ 216

9. 레드-블랙 트리(red-black tree)에 대한 설명으로 옳은 것만을 <보기>에서 모두 고른 것은?

<보기>

ㄱ. 루트 노드는 항상 검은색이다.
ㄴ. 검은 노드는 검은 노드를 자식으로 가질 수 없다.
ㄷ. 어떤 노드에서 리프(NIL)까지 가는 모든 경로에는 같은 수의 빨간 노드가 존재한다.
ㄹ. 레드-블랙 트리는 완전 이진 트리(complete binary tree)이다.

- ① ㄱ ② ㄴ, ㄷ
③ ㄱ, ㄷ, ㄹ ④ ㄱ, ㄴ, ㄷ, ㄹ

10. <보기>는 두 문자열 X 와 Y 가 주어졌을 때 최장 공통 부분 수열(LCS, longest common subsequence)의 길이를 구하는 알고리즘의 점화식이다. $c[i, j]$ 는 문자열 X_i 와 Y_j 의 LCS 길이라고 정의할 때, (가), (나)에 들어갈 내용으로 가장 옳은 것은? (단, LCS는 두 문자열에 공통으로 존재하는 가장 긴 부분 수열을 뜻한다. 이때, 문자의 연속성은 보장하지 않으며 순서는 유지하여야 한다.)

<보기>

$$c[i, j] = \begin{cases} 0 & \text{if } i=0 \text{ or } j=0 \\ \text{(가)} & \text{if } i, j > 0 \text{ and } x[i] = y[j] \\ \text{(나)} & \text{if } i, j > 0 \text{ and } x[i] \neq y[j] \end{cases}$$

- (가) (나)
- ① $c[i-1, j] + 1$ $\max\{c[i-1, j], c[i, j-1]\}$
② $c[i, j-1] + 1$ $\max\{c[i-1, j], c[i, j-1]\} + 1$
③ $c[i-1, j-1] + 1$ $\max\{c[i-1, j], c[i, j-1]\}$
④ $c[i-1, j-1] + 1$ $\max\{c[i-1, j], c[i, j-1]\} + 1$

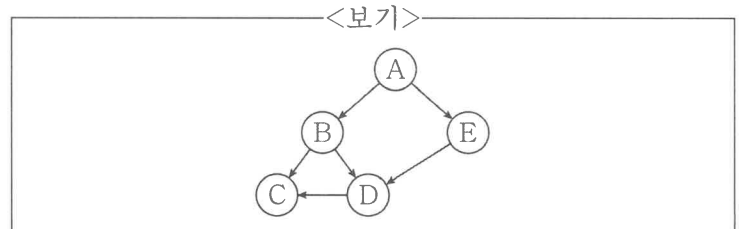
11. <보기>는 정수 n 을 입력하면 계산 결과를 출력(return)하는 알고리즘의 의사코드(pseudo code)이다. $f(n)$ 을 호출했을 때, 출력되는 값이 100보다 크게 되는 정수 n 의 최솟값은?

<보기>

```
f(n)
if (n < 1) return 1
else return f(n-1) + f(n-3) + 1
```

- ① 9 ② 10
③ 11 ④ 12

12. <보기> 그래프의 위상정렬(topological sort)의 결과로 가장 옳은 것은?



- ① $A \rightarrow B \rightarrow E \rightarrow D \rightarrow C$
② $A \rightarrow E \rightarrow D \rightarrow C \rightarrow B$
③ $A \rightarrow B \rightarrow C \rightarrow E \rightarrow D$
④ $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$

13. 이진 탐색 트리(BST, binary search tree)에 대한 설명으로 가장 옳지 않은 것은?

- ① 중복 키를 허용하지 않는 경우, 모든 노드에 대해 왼쪽 서브트리의 키는 해당 노드의 키보다 작고, 오른쪽 서브트리의 키는 해당 노드의 키보다 크다.
② 루트 노드는 항상 전체 원소의 중앙값이어야 한다.
③ BST에서 중위 순회(inorder traversal)를 하면 키 값이 오름차순으로 출력된다.
④ 탐색 · 삽입 · 삭제 연산의 시간복잡도는 트리의 높이에 비례한다.

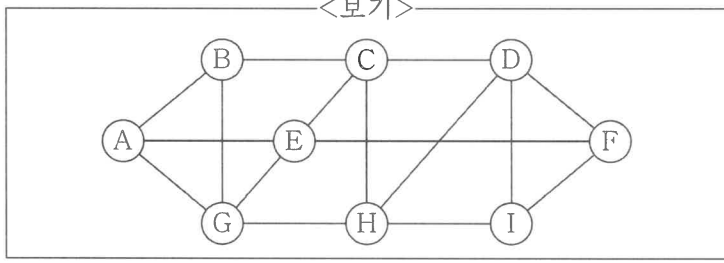
14. <보기>는 문자열 탐색 알고리즘인 라빈-카프(Rabin-Karp) 알고리즘에 대한 설명이다. 옳은 설명을 모두 고른 것은?

<보기>

ㄱ. 문자열 탐색을 위해 해시 함수를 사용한다.
ㄴ. 알고리즘의 효율성은 해시 함수 계산 비용을 줄이는 데에 있다.
ㄷ. 패턴의 해시값과 본문 부분 문자열의 해시값이 일치 하더라도 실제 문자가 일치하는지 추가 확인이 필요하다.

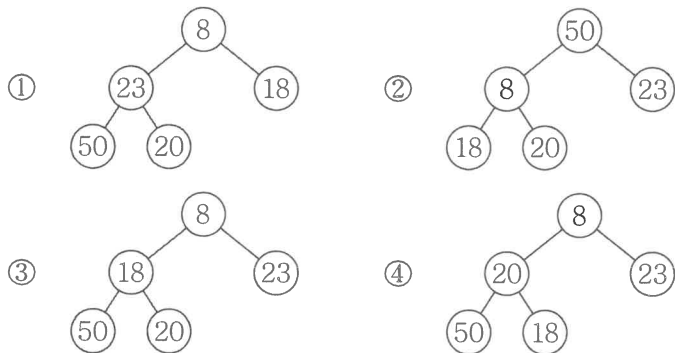
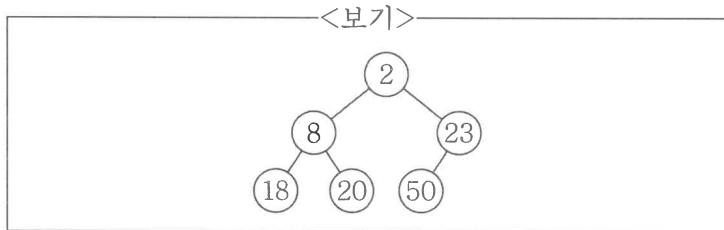
- ① ㄱ ② ㄱ, ㄴ
③ ㄴ, ㄷ ④ ㄱ, ㄴ, ㄷ

15. <보기>의 그래프에서 A부터 너비 우선 탐색(BFS, breadth first search)을 수행하는 경우, 정점의 방문 순서로 가장 옳은 것은? (단, 탐색되지 않은 이웃(neighbor)이 2개 이상 있는 경우 알파벳 순서로 탐색한다.)



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

16. <보기>의 최소 힙(min heap)에서 루트 노드를 삭제한 경우, 최소 힙 조건을 만족하도록 연산을 거친 후의 구조로 가장 옳은 것은?



17. 해시 테이블의 충돌 문제 해결을 위한 방법 중 개방 주소법(open addressing)에 해당하지 않는 것은?

- ① 선형 조사법(linear probing)
 ② 체이닝(chaining)
 ③ 이차 조사법(quadratic probing)
 ④ 이중 해싱(double hashing)

18. <보기>는 막대 자르기(rod cutting) 문제의 재귀적 의사코드(pseudo code)이다. 막대의 길이 정수 n 과 가격 배열 p 를 입력 받아, 막대를 잘랐을 때 가능한 이익의 최댓값을 구한다. 이때, <가>에 들어갈 내용으로 가장 옳은 것은? (단, 배열 p 의 길이는 n 이며, $p[i]$ 는 길이가 i 인 막대의 가격이다.)

<보기>

```

CUT_ROD(p, n)
    if n==0
        return 0
    q=-∞
    for i=1 to n
        q=max{q, p[i]+ _____ }
    return q
    
```

- ① CUT_ROD(p, i-1)
 ② CUT_ROD(p, i)
 ③ CUT_ROD(p, i+1)
 ④ CUT_ROD(p, n-i)

19. 정렬 알고리즘에 대한 설명 중 가장 옳지 않은 것은?

- ① 선택 정렬(selection sort) 알고리즘은 n 개의 원소를 갖는 리스트를 정렬하는 경우, 총 $n(n-1)/2$ 회의 비교연산이 진행된다.
 ② 힙 정렬(heap sort) 알고리즘은 완전 이진 트리 형태의 자료구조 힙을 활용한다.
 ③ 합병 정렬(merge sort) 알고리즘은 동적 계획법(dynamic programming) 방식을 기반으로 동작한다.
 ④ 기수 정렬(radix sort) 알고리즘은 자릿수가 d 인 자연수 n 개로 구성된 리스트를 정렬할 때, $d=n$ 이라면 $O(n^2)$ 의 시간복잡도를 갖는다.

20. <보기>에서 설명하는 문제 부류에 속하지 않는 것으로 가장 옳은 것은?

<보기>

- 현재 이 부류 문제에 대한 다항 시간의 알고리즘이 알려져 있지 않다.
- 이 부류의 문제 사이에는 다항 시간 환원(reduction)이 가능하다.
- 이 부류에 속하는 문제 중 어느 한 문제만 다항식 시간에 해결되면, 모든 다른 문제들이 다항 시간에 해결 가능하다.

- ① 만족성(SAT) 문제
 ② 부분 집합의 합(subset sum) 문제
 ③ 0-1 배낭 결정(knapsack decision) 문제
 ④ 모든 쌍 최단 경로(all pairs shortest paths) 문제