

[1-1] 각 수식에 해당하는 점근적 표기를 보기에서 모두 고르세요. 수학적으로 해당하는 것을 모두 적을 것

보기:

$O(n)$	$O(n^2)$	$\Theta(\log n)$	$\Theta(n^2)$	$\Omega(n)$	$\Omega(n^2)$
--------	----------	------------------	---------------	-------------	---------------

- (1) $10n^2 + 10n$
- (2) $3n + \log n + 3$

[2-1] 다음 5개의 수식은 문제의 크기가 n 일 때, 알고리즘의 수행시간 $T(n)$ 을 나열한 것이다. 물음에 답하세요.

$5n^2 + n$

$n + \log n$

$n \log n$

10

2^n

- (1) 이 수식들 중에서 $\Omega(n^2)$ 인 것을 모두 고르세요. 수학적으로 해당하는 것을 모두 적을 것
- (2) 이 수식들을 복잡도가 낮은 것부터(수행시간 면에서 효율적인 것부터) 차례대로 나열하세요. 복잡도가 같은 경우, 아무 수식이나 먼저 적어도 됨

[1-2] 문제의 크기가 n 일 때 다음 알고리즘의 수행시간 복잡도를 점근적 표기로 나타내세요.

```
algorithm(n) {  
  
    for(int i = 0; i < n; i++)  
        for(int j = 0; j < n; j++)  
            for(int k = 0; k < n; k++)  
                System.out.println(n);  
  
    for(int i = 0; i < n; i++)  
        for(int j = 0; j < n; j++)  
            System.out.println(n);  
}
```

[2-2] 문제의 크기가 n 일 때 다음 알고리즘의 수행시간 $T(n)$ 을 마스터 정리를 이용하여 점근적 표기법(asymptotic notation)으로 나타내세요. 풀이 과정과 답을 적을 것

```
algorithm(n) {  
    if(n > 1) {  
        System.out.println(n);  
        algorithm(n/2);  
        algorithm(n/2);  
    }  
}
```

과정:

답:

[1-3] 다음과 같은 계수 정렬(counting sort) 알고리즘을 이용하여 정렬하고자 한다. 물음에 답하세요.

```
countingSort(A, B, n)  ▷ A: 입력배열, B:정렬결과 n: 입력크기, 배열원소는 1~k 범위의 정수
{
    for i ← 1 to k      ▷ 첫번째 반복문
        C[i] ← 0;
    for j ← 1 to n      ▷ 두번째 반복문
        C[A[j]]++;
    for i ← 2 to k      ▷ 세번째 반복문
        C[i] ← C[i] + C[i-1];
    for j ← n downto 1 { ▷ 네번째 반복문
        B[C[A[j]]] ← A[j];
        C[A[j]]--;
    }
}
```

(1) 다음과 같은 1~4 범위의 정수값 10개를 오름차순 정렬하는 경우, 위 알고리즘의 for 반복문 4개 중에서, 세번째 반복문까지 수행을 마친 후 배열 C의 원소 값을 모두 적으세요.

4 1 4 3 2 4 1 2 1 4

(2) 네번째 반복문에서 n부터 1까지 역순으로 배열 B에 저장하는 이유는 무엇인가?

(3) 1~4 범위의 정수값 n개를 정렬할 때, 이 정렬 알고리즘의 수행시간 복잡도를 점근적 표기로 나타내세요.

[2-3] 다음은 정렬에 관한 문제이다. 각 문제의 답을 보기에서 모두 골라 적으세요. 단, n 은 정렬할 원소 수이다.

보기:

radix sort, selection sort, bubble sort,
quick sort, heap sort, counting sort

- (1) 평균 수행시간이 $O(n^2)$ 인 것은?
- (2) 평균 수행시간이 $O(n \log n)$ 인 것은?
- (3) 원소들끼리 키값을 비교하지 않고 정렬을 수행하는 것은?
- (4) 안정 정렬인 것은?

[2-4] 아래와 같은 선택 알고리즘 select를 다음과 같은 배열 A에 대해 수행하고자 한다. 물음에 답하세요.

A

0	1	2	3	4	5	6	7	8	9
1	3	5	8	9	2	10	7	6	4

```
select(A[], p, r, i) ▷ 배열 A[p...r]에서 i번째 작은 원소를 찾는다.(1번째 작은 원소가 최소 원소)
{
    if (p = r) then return A[p] ;

    q ← partition(A, p, r) ; ▷ A[r]을 기준으로 배열 A[p...r]를 분할하여 기준값 위치가 q

    k ← q-p+1;
    if (i < k) then return select(A, p, q-1, i) ; // (가)
    else if (i = k) then return A[q] ;
    else return select(A, q+1, r, i-k) ;           // (나)
}
```

(1) select(A, 0, 9, 8)이 호출되었을 때, (가), (나) 중에서 어떤 문장이 수행되는가? 그리고 그 문장의 select() 재귀 호출에서 매개변수 값은 무엇인가?

수행 문장 : (가), (나) 중 하나를 적을 것
매개변수: 수행 문장에서 select(A, __, __, __); 의 __ 부분을 숫자로 채우면 됨

(2) 원소 수가 n일 때, select 알고리즘의 평균 수행시간 복잡도를 점근적 표기로 나타내세요.

[1-5] 레드-블랙 트리에 관한 다음 물음에 답하세요.

(1) 비어 있는 레드-블랙 트리에 3, 4, 2, 7, 6, 5를 차례대로 삽입한 후 트리 상태를 그림으로 나타내세요.

- 단색으로 표시하려면 레드 노드는 키값만 적고, 블랙 노드는 원 안에 키값을 적을 것
- NIL 노드는 표시하지 않아도 됨

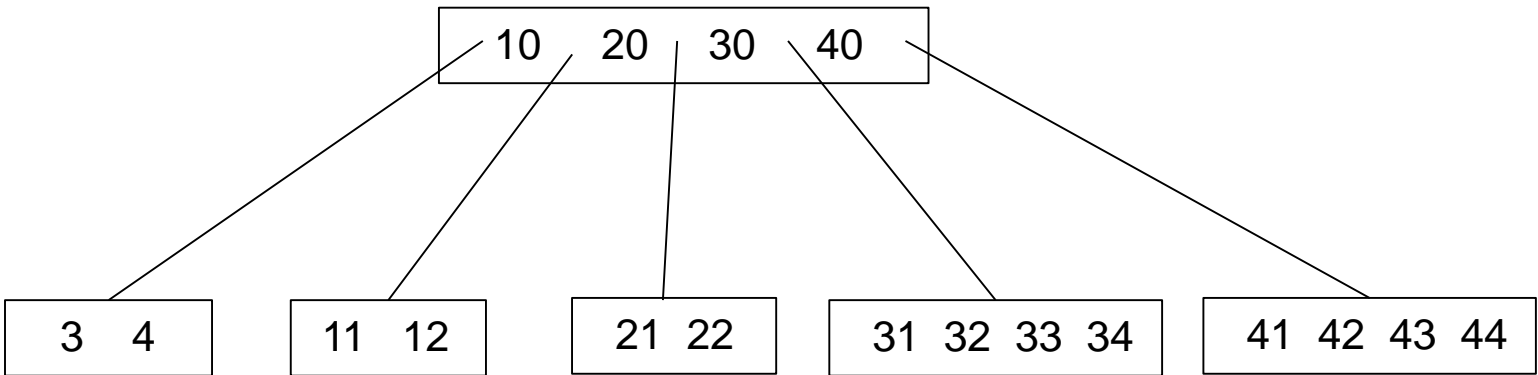
삽입 과정:

최종 트리 상태:

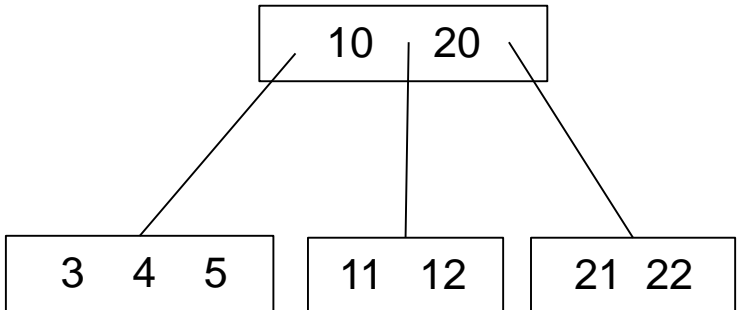
(2) 괄호 안에서 올바른 것을 고르세요. 레드-블랙 트리의 노드 수가 n 일 때, 하나의 키값 검색 시간은 최악의 경우 ($O(n^2)$, $O(\log n)$, $O(n)$, $O(n \log n)$)이다.

[2-5] 노드 당 최대 키 개수가 4인 B-트리에 대한 다음 물음에 답하세요. B-트리 성질만 유지하도록 연산을 수행하면 안되고, 수업 시간에 배운 알고리즘을 적용해야 함

(1) 다음 B-트리에 50을 삽입한 후 트리 상태를 그림으로 나타내세요.



(2) 다음 B-트리에서 11을 삭제한 후 트리 상태를 그림으로 나타내세요.



(3) 괄호 안에서 올바른 것을 고르세요. 저장된 키 값이 n개일 때, 하나의 키값 검색 시간은 평균적으로 ($O(n \log n)$, $O(n^2)$, $O(\log n)$, $O(n)$)이고, 최악의 경우 ($O(n \log n)$, $O(n^2)$, $O(\log n)$, $O(n)$)이다.

[1-6] 다음과 같은 해시 테이블에 대해 물음에 답하세요.

해시 함수 : 나누기 방법 $h(x) = x \bmod 11$

충돌 해결 : open addressing 중에서 이차원 조사(quadratic probing)

$$h_i(x) = (h(x) + i^2) \bmod 11, \quad i = 0, 1, 2, \dots$$

- (1) 해시 테이블의 상태가 오른쪽 그림과 같을 때, 적재율(load factor)은 얼마인가?
- (2) 괄호 안에서 올바른 것을 골라 적으세요. 해시 테이블의 적재율이 높을 수록 충돌 발생 가능성이 (높아진다, 낮아진다).
- (3) 괄호 안에서 올바른 것을 골라 적으세요. 해시 함수로 나누기 방법을 사용하므로 해시 테이블 크기를 2^k 로 하는 것이 (바람직하다. 바람직하지 않다).
- (4) 해시 테이블의 상태가 오른쪽 그림과 같을 때, 11을 검색하는 경우 몇 번의 조사가 일어나는가? (즉, 검사하는 해시 테이블 칸 개수는?)
- (5) 해시 테이블의 상태가 오른쪽 그림과 같을 때, 11, 22를 차례대로 삽입한 경우 해시 테이블의 상태를 그림으로 그리세요.

0	0
1	1
2	2
3	3
4	
5	
6	
7	
8	
9	
10	

[2-6] 다음과 같은 해시 테이블에 대해 물음에 답하세요.

해시 함수 : 나누기 방법 $h(x) = x \bmod 11$

충돌 해결 : open addressing 중에서 linear probing

- (1) 해시 테이블의 상태가 오른쪽 그림과 같을 때, 12를 검색하는 경우 조사 횟수는? (즉, 검사하는 해시 테이블 칸 개수는?)
- (2) 해시 테이블의 상태가 오른쪽 그림과 같을 때, 5, 12, 23을 차례대로 삽입한 후 해시 테이블의 상태를 그림으로 그리세요.
- (3) 이 해시 테이블에서 키값을 단순히 삭제하면 이후 다른 키 값 검색에서 문제가 생길 수 있다. 키 값 삭제를 어떤 방식으로 처리하면 이 문제를 해결할 수 있는가?
- (4) 이 해시 테이블의 문제점 중 한가지는 동일한 해시 주소를 갖지 않는 키값들 끼리도 서로 영향을 준다는 점이다. 해시 테이블 기법을 이용하되 이 문제점을 완전히 해결하기 위한 해법은 무엇인가? 해법을 길게 설명할 필요는 없음

0	
1	1
2	2
3	3
4	
5	
6	
7	
8	
9	
10	