

자료구조 및 알고리즘 중간시험 리포트

이름: _____
학번: _____
학과명: _____

본인은 본 시험을 어떠한 부정행위도 없이
치를 것을 서약합니다. (서명: _____)

제출마감: 2023.4/22 (일) 오후 10시

1. 다음의 알고리즘 SAMPLE을 참조하라.

```
SAMPLE(int n)
    int sum = 0;
    for (i = n; i > 0; i = i/2) do
        for (j = 0; j < n; j = j+1) do
            sum = sum + 1
    return(sum)
```

(①)
(②)
(③)
(④)
(⑤)

(1) 위의 각 괄호와 일치하는 왼쪽의 명령어들이 총 몇 번 수행하는지 모두 채워라.

(2) 위 알고리즘의 Worst Case의 시간 성능을 Big Oh 표기로 나타내라.

2. 다음의 알고리즘 SAMPLE을 참조하라. 여기서 X와 Y는 각각 n 개의 정수들을 저장하는 array이다.

```
SAMPLE(int (X[n])
    for (i = 0; i <= n-1; i = i+1) do
    {
        sum = X[0]
        for (j = 1; j <= i; j = j+1) do
            sum = sum + X[j]
        Y[i] = sum / (i + 1)
    }
    return Y
```

(1) X[8]에 다음 값이 있다고 할 때, 수행 후 Y[8]에 어떤 값들이 return되는지 보여라.

	0	1	2	3	4	5	6	7
X[8]	30	28	17	37	48	20	30	62

2. (2) 위의 알고리즘의 수행시간은 $O(n^2)$ 이다. 불필요한 연산을 소거하여 $O(n)$ 이 되도록 위와 동등한 알고리즘으로 작성하라.

3. (1) 다음은 10진수 n 을 2진수로 변환하는 알고리즘이다. 괄호 ①과 ②에 올바른 연산을 채워 완성하라.

```
CONVERT(int n)
    if n == 0 return 0;
    else { CONVERT( ① )
            return ( ② )
        }
```

(2) 다음은 정수 n 이 소수(prime number)이면 true를, 아니면 false를 return하는 알고리즘이다. 이 알고리즘의 시간 성능이 $O(\text{Root}(n))$ 이 되도록, 괄호 ①과 ②를 올바른 연산을 채워 완성하라.

```
TEST(int n)
    for (i = 2; ( ① ) <= n; i++)
        if ( ② ) == 0 return false
        else return true;
```

4. 다음은 Circular Queue의 Insert와 Delete 연산을 작성한 알고리즘이다.

```
max = 3; int queue[3];  
int rear = 0; int front = 0;
```

INSERT(int X)

```
{ rear = (rear + 1) MOD max  
  if (front == rear)  
  {  
    print("Sorry, queue is full")  
    ( ① )  
  }  
  print("OK, X is added")  
  queue[rear] = X  
}
```

DELETE

```
{ if ( ② ) print("Sorry, queue is empty")  
  front = (front + 1) MOD max  
  print("OK, X is deleted")  
  X = queue[front]  
}
```

1) 위의 알고리즘의 괄호 ①, ② 안에 들어 갈 명령어를 각각 작성하라.

2) 이제 아래와 같은 순서로 호출했다고 할 때, 출력되는 메시지를 차례대로 적어라.

1) INSERT(20); 2) INSERT(30); 3) INSERT(40); 4) INSERT(50);

5) DELETE; 6) INSERT(60); 7) DELETE; 8) DELETE; 9) DELETE

5. 다음의 15 개의 key들에 대해 Binary Search를 적용했을 때 3 번 이하의 비교로 찾을 수 있는 key들을 모두 적어라.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

6. 어떤 Binary Search Tree에 1부터 1000까지의 정수들이 저장되었다고 하자. 이제 363 이란 값을 탐색한다고 할 때, 다음 문항들 중 올바르지 않은 탐색 순서들에 해당하는 것을 찾아서 모두 적어라. 단 여러 개의 답안도 가능함.

1) 2, 252, 401, 398, 330, 344, 397, 363.

2) 924, 220, 911, 244, 898, 258, 362, 363.

3) 925, 202, 911, 240, 912, 245, 363.

4) 2, 399, 387, 219, 266, 382, 381, 278, 363.

5) 935, 278, 347, 621, 299, 392, 358, 363.

7. 다음의 후위 표현(Postfix)을 계산하는 과정을 단계별로 보여라.

여기서 %는 나머지 연산자이고, eos는 이 표현의 맨 끝을 나타내는 특수 심볼이다.

3 5 + 4 * 8 - 8 2 / 5 3 % + - eos

- 8.** (1) 높이가 h 인 Complete Binary Tree에서 node의 최소 개수와 최대 개수를 계산하여 각각 h 에 대한 식으로 표현하라. 단, root의 level을 1로 가정한다.
- (2) (1)에서 구한 node의 수를 n 이라 할 때, 이를 이용하여 Complete Binary Tree의 최대 높이를 n 으로 표현하라.

9. 다음의 비어 있는 array에 {30, 10, 40, 20, 70, 50, 60, 80} 의 8 개의 key들을 차례대로 insert 할 때, Max Heap이 만들어지는 과정을 완성시켜라. (array의 첫 주소는 1로 시작하는 것으로 하고, 반드시 각 과정을 array에서 보일 것.

1	2	3	4	5	6	7	8

10. 다음의 array에 구현된 Max Heap을 이용하여 8 개의 key들을 올림 차순으로 sort 하는 과정을 완성시켜라. (힌트: 매번 delete를 하면서, 삭제된 값을 array의 맨 끝에 저장할 것.)

1	2	3	4	5	6	7	8
30	25	28	20	18	12	16	15

Delete
→

1	2	3	4	5	6	7	8
28	25	16	20	18	12	15	30

Delete
→

11. 다음 6 개의 key들에 대해 다음의 각 순서로 삽입된 경우, 탐색 시간이 가장 작은 것을 골라라.

{2, 7, 8, 10, 15, 20}

1) 15, 2, 20, 8, 7

2) 8, 20, 7, 2, 15, 10

3) 7, 2, 10, 8, 15, 20

4) 10, 7, 15, 20, 8

12. 1 부터 10까지 10개의 정수들이 있다고 하자. 이제 Max Heap을 만들려고 하는데, PREORDER로 순회한 결과, 10 개의 정수들이 내림 차순으로 정렬된 결과를 보여줄 수 있는 Max Heap를 찾아 그려라.

13. n 개의 node를 갖는 Binary Tree에서 Leaf Node의 최소 개수와 최대 개수를 각각 구하라. 풀이 과정에서 최소와 최대인 경우에 Binary Tree가 각각 어느 유형인지를 보인 후, 이를 근거를 n 에 대한 수식으로 개수를 계산하라.