

연습문제 풀이를 마친 학생은 답안지를 제출하시기 바랍니다. 답안지를 제출한 후에는 퇴실할 수 있습니다.

Name: \_\_\_\_\_ Student ID: \_\_\_\_\_ Class: \_\_\_\_\_

담당교수: 김종규

---

1. Big-O 와 관련한 다음 질문에 답하시오.

- (a) 알고리즘의 복잡도가  $O(1)$ ,  $O(n \log n)$ ,  $O(n)$ ,  $O(n^2)$ ,  $O(\log n)$ ,  $O(2^n)$ ,  $O(n!)$  과 같이 주어져 있을 때, 알고리즘의 복잡도를 낮은 것에서 높은 순으로 나열하시오.

Answer:

$O(1)$ ,  $O(\log n)$ ,  $O(n)$ ,  $O(n \log n)$ ,  $O(n^2)$ ,  $O(2^n)$ ,  $O(n!)$

- (b) 다음에서 {some\_statements} 가 수행되는 횟수는? Big-O 로 나타내시오.

```
i = 1
while i <= n:
    j = 1
    while j <= i:
        {some_statements}
        j++
    i++
```

Answer:

outer loop  $O(n)$ , inner loop  $O(n)$ ,  $\therefore O(n^2)$

- (c) 다음에서 {some\_other\_statements} 가 수행되는 횟수는? Big-O 로 나타내시오.

```
q = 3;
for (i = n; i > q; i = (i + q) / 2) {
    for(j = 1; j <= n; j++) {
```

```

        {some_other_statements}
    }
}

```

Answer:

outer loop  $O(\log n)$ , inner loop  $O(n)$ ,  $\therefore O(n \log n)$

2. 다음은 배열 A 에 nelem 개의 값이 들어 있을 때 이를 insertion sort 알고리즘을 이용하여 정렬하는 프로그램이다. 작은 값에서 큰 값으로 정렬이 완성되도록 빈 칸에 적절한 값을 넣으시오.

```

void insertion_sort(int A[], int nelem)
{
    for (int j = 1; j < nelem; j++) { /* blank */
        int key = A[j];
        int i = ____;
        while (____) { /* blank */
            ____; /* blank */
            ____; /* blank */
        }
        ____ = key; /* blank */
    }
}

```

Answer:

```

void insertion_sort(int A[], int nelem)
{
    for (int j = 1; j < nelem; j++) {
        int key = A[j];
        int i = j - 1;
        while (i >= 0 && A[i] > key) {
            A[i + 1] = A[i];
            i = i - 1;
        }
    }
}

```

```
    A[i + 1] = key;
}
}
```

3. (5 points) 다음은 stack 의 push operation 이다. array index 가 0 에서 시작할 때 pop() operation 을 정의하시오.

```
void push(int val) {
    stk[top] = val;
    top = top + 1;
}
```

**Answer:**

```
int pop() {
    return stk[--top];
}
```

혹은

```
int pop() {
    top = top - 1;
    return stk[top];
}
```