

풀이시간은 30 분이며 사정에 따라 최대 10 분간 추가 시간을 줄 수 있습니다.  
문제 풀이를 마친 학생은 답안지를 제출하시기 바랍니다. 답안지를 제출한  
후에는 퇴실할 수 있습니다.

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

담당교수: 김종규

---

1. (10 points) 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  $\{s1\}$  과  $\{s2\}$  가 수행되는 횟수의 합은? Big-O 로 나타내시오.

```
function find(A,k):  
    # A is an array of [1..n]  
    # k is a value  
    p = 1  
    q = n  
    while p < q:  
        n = (p + q) / 2    # where n is an integer  
        if k < A[n]:  
            q = n - 1  
            {s1}  
        elif k > A[n]:  
            p = n + 1  
            {s2}  
    else  
        return n
```

```
    end while
end function
```

**Answer:**

$O(\log n)$

2. (10 points) 다음은 스택에 대한 질문입니다. 잘 읽고 물음에 답하시오.

- (a) 다음과 같이 array `stk[0..n-1]` 를 이용하여 stack 의 push operation 을 정의했다고 가정하자. 만일 stack 에 저장할 수 있는 이상으로 push 연산이 호출되면 `error("overflow")` 를 호출하여 프로그램의 수행을 종료하고자 한다. 빈 칸에 적합한 식을 넣으시오.

```
void push(int val) {
    if (_____) {
        error("overflow");
    }
    stk[top] = val;
    top = top + 1;
}
```

**Answer:**

```
void push(int val) {
    if (top >= n) {
        error("overflow")
    }
    stk[top] = val;
    top = top + 1;
}
```

- (b) 두 개의 stack `s1, s2` 를 이용하여 다음과 같은 연산을 정의하였다.

```
def p(item):
    push(s1, item)
def q():
    if (empty(s2)):
        while not empty(s1):
            push(s2, pop(s1))
    return pop(s2)
```

`push()` 와 `pop()` 이 모두  $O(1)$  의 시간이 소요된다고 가정하고 다음 프로그램의 복잡도를 Big- $O$  로 나타내시오.

```

1:  i = 0
2:  while i < n:
3:      p(i)
4:      i++
5:  q()
6:  q()
7:  q()

```

Answer:

line 2:  $O(n)$

line 5:  $O(n)$

line 6:  $O(1)$

$\therefore$  The complexity is  $O(n)$

3. (10 points) 다음은 그림 1 와 같이 sentinel 이 있는 doubly linked list 에 대한 질문입니다. 이 linked list 의 가장 앞에 새로운 element  $x$  를 넣는 알고리즘 List-Insert 가 그림 2 와 같이 주어 져 있을 때 이 리스트의 가장 마지막에  $x$  를 추가하는 알고리즘 List-Append 를 작성하시오.

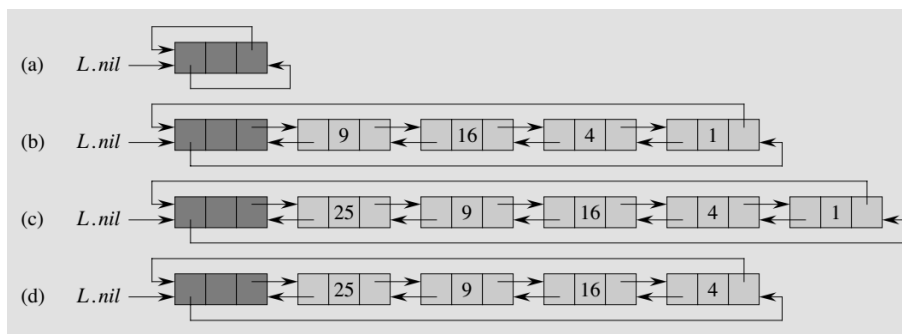


그림 1: Linked list with a sentinel

Answer:

```

List-Append(L, x)
    x.prev = L.nil.prev
    L.nil.prev.next = x

```

```
LIST-INSERT'(L, x)
1  x.next = L.nil.next
2  L.nil.next.prev = x
3  L.nil.next = x
4  x.prev = L.nil
```

그림 2: Inserting with a sentinel

```
L.nil.prev = x
x.next = L.nil
```