

## 미니퀴즈 (주관식) — 힙

날짜: 2025-10-17 이름: \_\_\_\_\_ 학번: \_\_\_\_\_

※ 간단명료하게 쓰세요. 풀이가 필요한 경우 핵심만 적기.

```
1. void min_heapify_all(struct min_heap *heap,
    const struct min_heap_callbacks *funcs)
```

{

---

---

---

---

}

```
2. void min_heap_sift_down(
    struct min_heap *heap, int pos,
    const struct min_heap_callbacks *funcs)
```

[illegible]

}

```
3. void min_heap_pop(struct min_heap *heap,
    const struct min_heap_callbacks *funcs)
```

{

---

---

---

---

---

---

---

---

---

---

}

```
4. void min_heap_push(struct min_heap *heap,
    int val,
    const struct min_heap_callbacks *funcs)
```

[illegible]

---

---

```
}
```

5. 다음에 주어진 code 들에 대해 time complexity 를 big-oh (O) notation 을 사용하여 표현하시오. 그리고, 그렇게 판단하는 이유도 함께 기술하시오.

5 - (1). int f1(int n)

```
{  
    int result = n * n;  
    return result;  
}
```

---

---

5 - (2). int f2(int n)

```
{  
    int result = 0;  
    while (n > 1) {  
        n = n / 2;  
        result += 1;  
    }  
    return result;  
}
```

---

---

5 - (3). int f3(int n, int m)

```
{  
    int result = 0;  
    int i, j;  
    for (i = 1; i < n; i *= 2)  
        result += 1;  
    for (j = 1; j < m; j *= 5)  
        result += 1;  
    return result;  
}
```

---

---

6. 크기가 6 인 circular queue 에 다음과 같이 삽입과 삭제가 순서대로 진행되었을 경우, 각 단계에서의 circular queue 의 내용을 적으시오.

단, front 와 rear 는 초기값으로 0 을 부여한다.

enqueue(1), enqueue(2), enqueue(3), enqueue(7),  
enqueue(6), enqueue(5), dequeue(), dequeue(),  
enqueue(4)

---

---

---

---

---

---

---

---

7. 다음과 같이 infix notation 을 postfix notation 으로 변경할 수 있다.

$3+4*2-7 \rightarrow 342*+7-$        $2*(4+6)-9 \rightarrow 246+*9-$

이러한 기능을 갖는 function 을 개발한다고 할 때, 그 function 의 알고리즘을 기술하시오. Flow chart 의 형태로 표현해도 좋고, pseudo code, 혹은 논리적으로 명확하고 간단한 우리글도 좋다. 단, 알고리즘대로 절차를 수행했을 때 변환이 완료될 수 있어야 한다.

---

---

---

---

---

---

---

---

8. 아래 프로그램에서 ①, ②, ③을 수행하고 난 후 head 가 가리키는 linked list 가 어떤 모양인지 그림으로 나타내시오.

```
#include <stdio.h>
#include <stdlib.h>

typedef struct node* nodePointer;
typedef struct node {
    int data;
    nodePointer prev;
    nodePointer next;
} node;

nodePointer GetNewNode(int x) {
    nodePointer newNode = malloc(sizeof(node));
    newNode->data = x;
    newNode->prev = NULL;
    newNode->next = NULL;
    return newNode;
}

void f1(int x, nodePointer *ptr)
{
    nodePointer newNode = GetNewNode(x);
    if (*ptr == NULL) {
        *ptr = newNode;
        return;
    }
    (*ptr)->prev = newNode;
    newNode->next = *ptr;
    *ptr = newNode;
}

void f2(int x, nodePointer *ptr)
{
    nodePointer temp = *ptr;
    nodePointer newNode = GetNewNode(x);
    if (*ptr == NULL) {
        *ptr = newNode;
        return;
    }
    while (temp->next != NULL)
        temp = temp->next;
```

```
temp->next = newNode;
newNode->prev = temp;
}

int main()
{
    nodePointer head = NULL;

    f1(2, &head);    // ①
    f2(3, &head);    // ②
    f1(4, &head);    // ③
    return 0;
}
```

9. Node 의 개수가 n 일 때 최소 height 와 최대 height 를 갖는 경우, 그 값이 얼마인지 각각 예를 들어 설명하시오.

10. Node 의 개수가 n 일 때 최소 width 와 최대 width 를 갖는 경우, 그 값이 얼마인지 각각 예를 들어 설명하시오.

11. postorder traversal 결과가 D B F G E H C A, inorder traversal 결과가 B D A F E G C H 인 binary tree 를 그림으로 나타내시오.