

# 4week-BST 번역

## Section F – 이진 검색 트리 (Binary Search Trees) [이 섹션의 지정된 질문에 대한 답변을 작성하시오.]

**정보:** APAS 시스템에서 제공하는 프로그램 템플릿을 사용하여 질문에 대한 함수를 구현해야 한다.

1. **(levelOrderTraversal)** 반복적인 C 함수 `levelOrderTraversal` 을 작성하시오. 이 함수는 큐를 사용하여 루트 노드 레벨에서 시작하여 이진 트리를 레벨별로 순회하면서 출력한다. 이때 큐에서 정수를 추가하거나 제거할 때는 `enqueue()` 또는 `dequeue()` 연산만 사용해야 한다. 큐가 비어 있지 않으면, 시작할 때 큐를 비워야 한다.

함수 프로토타입은 다음과 같다:

```
void levelOrderIterative(BSTNode *root);
```

예를 들어, 아래 그림 1에 나오는 이진 트리의 경우, 레벨 순서 순회 결과는 `20, 15, 50, 10, 18, 25, 80` 이다.

2. **(inOrderIterative)** 반복적인 C 함수 `inOrderIterative()` 를 작성하시오. 이 함수는 스택을 사용하여 이진 검색 트리의 중위(in-order) 순회를 출력한다. 이때 스택에서 정수를 추가하거나 제거할 때는 `push()` 또는 `pop()` 연산만 사용해야 한다. 스택이 비어 있지 않으면, 시작할 때 스택을 비워야 한다.

함수 프로토타입은 다음과 같다:

```
void inOrderIterative(BSTNode *root);
```

예를 들어, 아래 그림 2에 나오는 이진 트리의 경우, 반복적인 중위 순회 결과는 `10, 15, 18, 20, 50` 이다.

3. **(preOrderIterative)** 반복적인 C 함수 `preOrderIterative()` 를 작성하시오. 이 함수는 스택을 사용하여 이진 검색 트리의 전위(pre-order) 순회를 출력한다. 이때 스택에서 정수를 추가하거나 제거할 때는 `push()` 또는 `pop()` 연산만 사용해야 한다. 스택이 비어 있지 않으면, 시작할 때 스택을 비워야 한다.

함수 프로토타입은 다음과 같다:

```
void preOrderIterative(BSTNode *root);
```

예를 들어, 아래 그림 3에 나오는 이진 트리의 경우, 반복적인 전위 순회 결과는 20, 15, 10, 18, 50, 25, 80 이다.

4. **(postOrderIterativeS1)** 반복적인 C 함수 `postOrderIterativeS1()` 을 작성하시오. 이 함수는 스택을 사용하여 이진 검색 트리의 후위(post-order) 순회를 출력한다. 이때 스택에서 정수를 추가하거나 제거할 때는 `push()` 또는 `pop()` 연산만 사용해야 한다. 스택이 비어 있지 않으면, 시작할 때 스택을 비워야 한다.

함수 프로토타입은 다음과 같다:

```
void postOrderIterativeS1(BSTNode *root);
```

예를 들어, 아래 그림 4에 나오는 이진 트리의 경우, 반복적인 후위 순회 결과는 10, 18, 15, 25, 80, 50, 20 이다.

5. **(postOrderIterativeS2)** 반복적인 C 함수 `postOrderIterativeS2()` 를 작성하시오. 이 함수는 두 개의 스택을 사용하여 이진 검색 트리의 후위 순회를 출력한다. 이때 스택에서 정수를 추가하거나 제거할 때는 `push()` 또는 `pop()` 연산만 사용해야 한다. 스택이 비어 있지 않으면, 시작할 때 스택을 비워야 한다.

함수 프로토타입은 다음과 같다:

```
void postOrderIterativeS2(BSTNode *root);
```

예를 들어, 아래 그림 5에 나오는 이진 트리의 경우, 반복적인 후위 순회 결과는 10, 18, 15, 25, 80, 50, 20 이다.