섹션 F - 이진 검색 트리

정보: 문제용 프로그램 템플릿은 APAS 시스템에서 사용할 수 있습니다. 이를 사용하여 함수를 구현해야 합니다.

1. **(levelOrderTraversal)** 루트 노드 수준에서 시작하여 **큐를** 사용하여 이진 트리의 수준별 순회를 출력하는 반복적 C 함수 `levelOrderTraversal`을 작성합니다. 큐에서 정수를 추가하거나 제거할 때만 `enqueue()` 또는 `dequeue()` 연산을 사용해야 한다는 점에 유의하세요. 큐가 비어 있지 않은 경우 처음에 큐를 비워야 한다는 점을 잊지 마세요.

   함수 프로토타입은 다음과 같습니다:

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

   예를 들어 *그림 1의* 이진 트리의 경우 레벨 순서 트리 탐색은 **20, 15, 50, 10, 18, 25, 80입니다**.

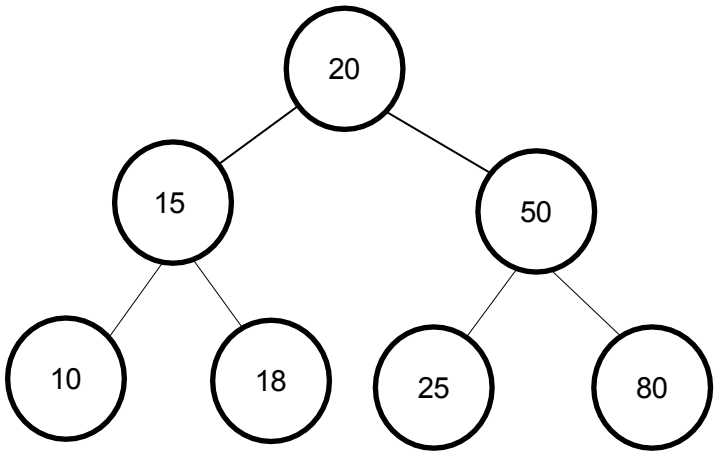


그림 1

2. **(inOrderIterative)** **스택을** 사용하여 이진 검색 트리의 순서대로 순회하는 것을 출력하는 반복적 C 함수 `inOrderIterative()`를 작성합니다. 스택에서 정수를 추가하거나 제거할 때만 `push()` 또는 `pop()` 연산을 사용해야 한다는 점에 유의하세요. 스택이 비어 있지 않은 경우 처음에 스택을 비워야 한다는 점을 잊지 마세요.

   함수 프로토타입은 다음과 같습니다:

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

   예를 들어, *그림 2의* 이진 트리의 경우 반복 순서 탐색은 다음과 같습니다: **10, 15, 18, 20, 50.**

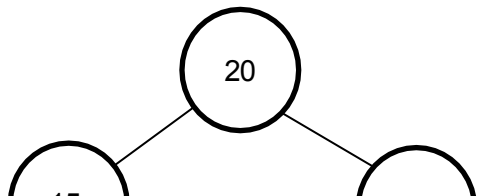


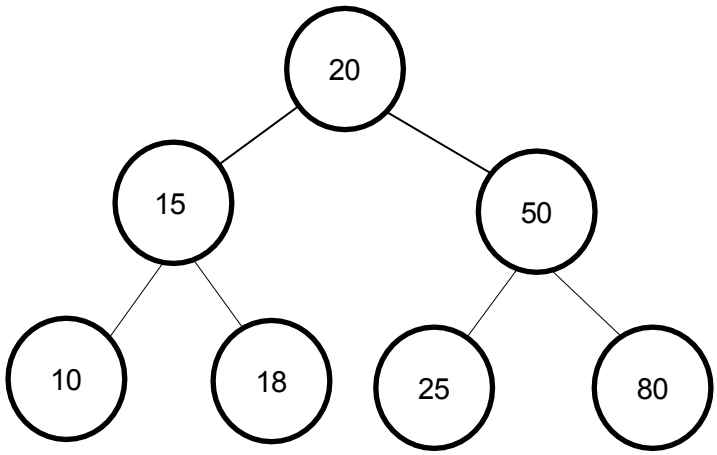
그림 2

3. **(preOrderIterative) 스택을** 사용하여 이진 검색 트리의 사전 순서 순회를 출력하는 반복적 C 함수 `preOrderIterative()`를 작성합니다. 스택에서 정수를 추가하거나 제거할 때만 `push()` 또는 `pop()` 연산을 사용해야 한다는 점에 유의하세요. 스택이 비어 있지 않은 경우 처음에 스택을 비워야 한다는 점을 잊지 마세요.

함수 프로토타입은 다음과 같습니다:

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

예를 들어 *그림 3의* 이진 트리의 경우 반복적인 프리오더 트리 탐색은 **20, 15, 10, 18, 50, 25, 80**



입니다.

그림 3

4. **(postOrderIterativeS1) 스택을** 사용하여 이진 검색 트리의 사후 순서를 출력하는 반복적 C 함수 `postOrderIterativeS1()`을 작성합니다. 스택에서 정수를 추가하거나 제거할 때만 `push()` 또는 `pop()` 연산을 사용해야 한다는 점에 유의하세요. 스택이 비어 있지 않은 경우 처음에 스택을 비워야 한다는 점을 잊지 마세요.

함수 프로토타입은 다음과 같습니다:

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

예를 들어, *그림 4의* 이진 트리의 경우 반복적인 포스트오더 트리 탐색은 다음과 같습니다: **10, 18, 15, 25, 80, 50, 20.**

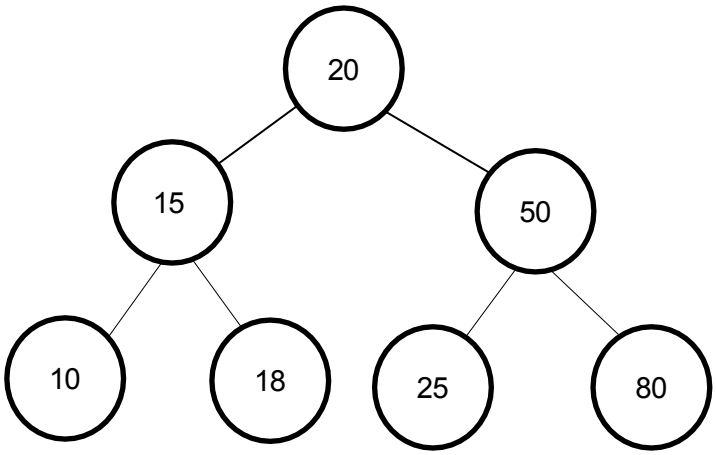


그림 4

5. **(postOrderIterativeS2) 두 개의 스택을** 사용하여 이진 검색 트리의 사후 순서를 출력하는 반복적 C 함수 `postOrderIterativeS2()`를 작성합니다. 스택에서 정수를 추가하거나 제거할 때만 `push()` 또는 `pop()` 연산을 사용해야 한다는 점에 유의하세요. 스택이 비어 있지 않은 경우 처음에 스택을 비우는 것을 잊지 마세요.

함수 프로토타입은 다음과 같습니다:

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

예를 들어, *그림 5의* 이진 트리의 경우 반복적인 포스트오더 트리 탐색은 다음과 같습니다: **10, 18, 15, 25, 80, 50, 20.**

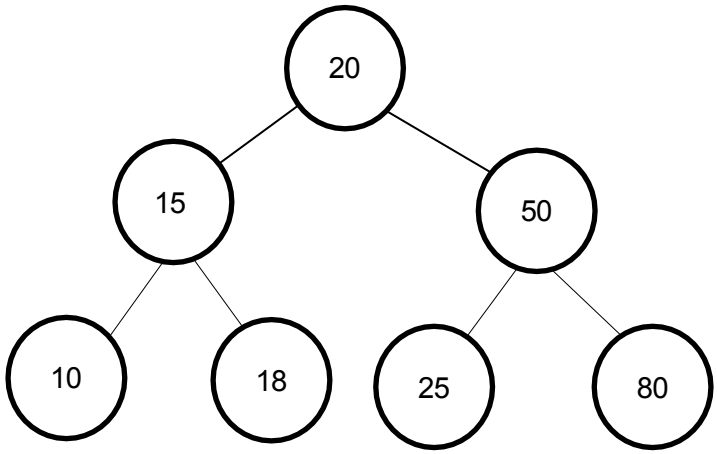


그림 5