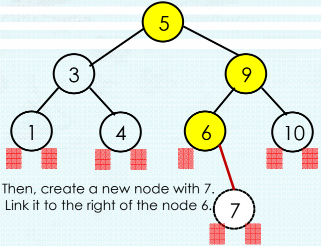
Operations: Insert (or grow)

- grow(node, k) - Insert a node with k
  - **Step 1**: If the tree is empty, return a new node(k).
  - **Step 2**: Pretending to search for k in BST, until locating a nullptr.
  - **Step 3**: create a new node(k) and link it.

```
tree grow(tree node, int key) {
  if (node == nullptr)
    return new tree(key);

  if (key < node->key)
    node->left = grow(node->left, key);
  else if (key > node->key)
    node->right = grow(node->right, key);
  return node;
}
```

- Q1: Explain the differences between the binary tree and binary search tree in this operation.
- -->binary tree는 key value가 들어오는 순서대로 tree에 집어넣지만, binary search tree는 자기 자신의 node를 기준으로 왼쪽 node에는 자기 자신 node의 key보다 작은 key를 가진 node를, 오른 쪽 node에는 자기 자신의 node의 key보다 큰 값을 가진 node를 집어 넣습니다.
- Q2: To complete inserting **7**, how many times was **`grow()`** called?
- -->3번
- Q3: How many times "**`if (key < node->key)`**" called during this process or inserting 7?
- -->1번
- Q4: At the end of this whole process, which **`return`** will be executed and what is the key value of the node?
- -->return node
- -->key value of node is 9.



Then, create a new node with 7.
Link it to the right of the node 6.