실습 9 – 이진 탐색 트리

- 실습 목표
  - 이진 탐색 트리의 알고리즘(생성, 검색)을 이해한다.
  - 이진 트리의 추가 연산들(노드 수, 높이, 단말 노드의 수)을 구현할 수 있다.

이진 탐색 트리의 생성

- struct node {int key; double value; *lchild; *rchild;}

- **(9.1.)** 프로그램 5.17과 강의 노트의 modified_search() 알고리즘을 구현
  - modified_search는 key를 이용하여 검색
  - 함수원형 :
    treePointer modified_search(treePointer tree, int key)

- main 함수에서는 12개의 (key, value) 쌍을 생성하여 이진 탐색 트리에 추가

- inorder 순회 알고리즘을 이용하여 이진 탐색 트리를 순회하면서 (key, value) 쌍을 출력

이진 탐색 트리의 검색

- **(9.2.)** key를 입력받아, 이에 해당하는 value를 찾는 search 알고리즘 구현 (프로그램 5.15 또는 프로그램 5.16)

- main에서 사용자에게 key를 입력받고, search 알고리즘을 호출하여 실행 결과를 확인

이진 트리의 추가 연산

- **(9.3.)** n을 인자로 받아, n개의
  - key: (rand() / (0x7fff * 1.0)) * 100,000,000
  - value: 1.0 / key
  - 쌍을 이진 탐색 트리에 추가하는 함수 A = make_bst(int n)을 작성
  - 함수원형 : treePointer make_bst(int n)
  - n의 크기를 다양하게 변경하면서 make_bst의 실행 시간을 관찰 (n: 100 ~ 1,000,000)

- make_bst() 함수로 자동 생성한 트리의 노드 수, 높이(깊이), 단말노드 수를 각각 출력
  - 함수원형 :
    int count_node(treePointer ptr)
    int count_depth(treePointer ptr)
    int count_leaf(treePointer ptr)

이진 트리의 추가 연산

- 이진 탐색 트리의 삭제 알고리즘을 구현한 후, 실행 결과를 확인