자료구조 (Data Structure)
13주차: AVL 트리

지난 시간 내용 - balance factor

- 균형의 정의
- 높이의 정의

지난 시간 내용 - Left-Left rotation

- 왼쪽으로 균형이 기울어졌을 때,
- 그리고, 왼쪽 자식의 왼쪽 서브트리가 오른쪽보다 낮지 않을 때

지난 시간 내용 - Left-Right rotation

- 왼쪽으로 균형이 기울어졌을 때,
- 그리고, 왼쪽 자식의 오른쪽 서브트리가 왼쪽보다 높을 때

지난 시간 내용 - Right-Right rotation

- 오른쪽으로 균형이 기울어졌을 때,
- 그리고, 오른쪽 자식의 오른쪽 서브트리가 왼쪽보다 낮지 않을 때

지난 시간 내용 - Right-Left rotation

- 오른쪽으로 균형이 기울어졌을 때,
- 그리고, 오른쪽 자식의 왼쪽 서브트리가 오른쪽보다 높을 때

지난 시간 내용 - 높이 계산

- post-order에서 계산한다
- 자식들의 높이 중 큰 값 + 1

지난 시간 내용 - 균형 회복

- post-order에서 균형 계산 후 필요하면 회복한다

오늘의 목차

- 이진탐색트리에서 노드 추가하기
- balance factor 구하기
- 균형 회복하기

노드 추가

함수 add_node_bst(노드 *root, 노드 *node):

- if (!root)
  root = node;
- if (!node || node->data == root->data)
  ;
- else if (node->data < root->data)
  root->left = add_node_bst(root->left, node);
- else
  root->right = add_node_bst(root->right, node);
- return root;

노드 추가 - 기출문제 2018

- bst를 새로 만들어서 다음 순서로 노드를 추가했을

경우, bst에 자식이 2개인 노드가 없는지를 답하시오

1) 데이터: 0, 6, 5, 1, 4, 3, 2
2) 데이터: 6, 0, 5, 1, 4, 2, 3
3) 데이터: 0, 6, 2, 5, 1, 4, 3
4) 데이터: 6, 0, 5, 4, 1, 2, 3

노드 추가 - 기출문제 2018

- bst를 새로 만들어서 다음 순서로 노드를 추가했을

경우, 트리가 만들어지는 과정을 그리시오

데이터: 5, 2, 0, 1, 4, 3, 7, 6, 8

노드 추가 - 기출문제 2021

- bst를 새로 만들어서 다음 순서로 노드를 추가했을

경우, 만들어진 트리를 그리시오

데이터: 1, 2, 3, 4, 5, 6, 7

balance factor

함수 get_height(노드 *n):

- return n ? n->height : 0;

함수 balance_factor(노드 *n):

- return get_height(n->left) - get_height(n->right);

balance factor - 높이 계산

함수 set_height(노드 *n):

- l_h = get_height(node->left);
- r_h = get_height(node->right);

- if (l_h > r_h)
  n->height = l_h + 1;
- else
  n->height = r_h + 1

balance factor - 전체 높이 계산

함수 set_height_all(노드 *root):

- if (!root) return;

- set_height_all(root->left);
- set_height_all(root->right);

- set_height(root);
- printf("node %d(bf: %d)\n", root->data, bf(root));

balance factor - 기출문제 2018

- bst를 새로 만들어서 다음 순서로 노드를 추가했을

경우, 각 노드의 balance factor를 적으시오

데이터: 3, 2, 6, 4, 7

균형 회복

함수 rebalance(노드 *n):

- bf = balance_factor(n);
- if (bf >= 2)

```
        노드 *l = node->left
        if (get_height(l->left) >= get_height(l->right))
                n = ll_rotate(n);
        else
                n = lr_rotate(n);
```

균형 회복

- else if (bf <= -2)

  노드 *r = node->right

  if (get_height(r->right) >= get_height(l->left))

    n = rr_rotate(n);

  else

    n = rl_rotate(n);

- return n;

균형 회복 - 기출문제 2018

- 새로운 bst에 다음 순서로 노드를 추가한 후에

루트에서 균형을 회복한 결과를 그리시오

데이터: 3, 2, 6, 4, 7, 5

균형 회복 - 기출문제 2018

- 새로운 bst에 다음 순서로 노드를 추가한 후에

루트에서 균형을 회복한 결과를 그리시오

데이터: 3, 2, 6, 4, 7, 8

균형 회복 - 기출문제 2018

- 새로운 bst에 다음 순서로 노드를 추가한 후에

루트의 왼쪽 자식에서 균형을 회복한 결과를 그리시오

데이터: 3, 2, 0, 1, 6, 4, 7

균형 회복 - 노드 추가

함수 add_node(노드 *root, 노드 *n):

- if (!root)

  root = n;

- else

  root = _add_node(root, n);

- set_height(root);
- return rebalance(root);

균형 회복 - 노드 추가

함수 _add_node(노드 *root, 노드 *n):

- if (!node || node->data == root->data)
  ;
- else if (node->data < root->data)
  root->left = add_node(root->left, n);
- else
  root->right = add_node(root->right, n);
- return root;

균형 회복 - 노드 추가 - 기출문제 2021

- 새로운 AVL tree에 다음 순서로 노드를 추가할 때

트리가 만들어지는 과정을 그리시오

데이터: 1, 2, 3, 4, 5, 6, 7

오늘 한 내용

- bst에서 새 노드의 적절한 자리를 찾아서 추가한다
- balance factor는 post_order에서
  왼쪽 height - 오른쪽 height로 구한다
- 노드를 추가할 때 적절히 회전을 해서 균형을 맞춘다