EDDI
Electronic Design
Development Institute

에디로봇아카데미

임베디드 마스터 Lv2 과정

제 1기

2021. 12. 29

손표훈

CONTENTS

EDDI
Electronic Design
Development Institute

- RB트리 삽입 검토
- RB트리 삭제 구현 전략
  - RB트리 삭제 전략 구성
  - Black 노드가 삭제되는 경우
    - Case 1 : 형제 = red
    - Case 2 : 형제 = black, 조카 = null or black
    - Case 3 : 형제 = black, 조카 = red

RB트리 삽입전략

EDDI
Electronic Design
Development Institute

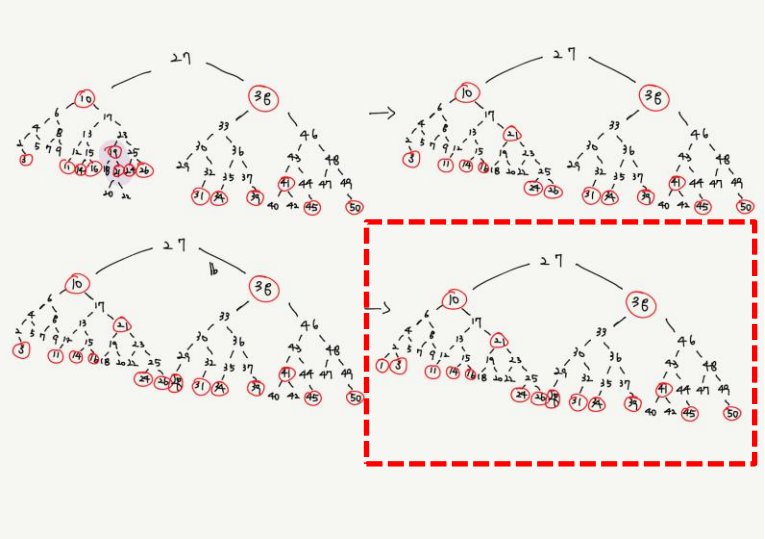
- RB 트리 삽입 검토
  - ➢ 삽입 TODO 리스트

```
/**************TODO**************/
/*
#top_root = main문의 root, root = 자식노드가 red인 노드
#지역변수 : me = 자식노드가 red인 노드, parent = me의 부모, brother = me의 형제, child = me의 자식
1. child 찾기
2. child 또는 parent가 NULL이면 return
3. brother 찾기
4. me와 child가 red인지 확인
        4.1 me와 child가 red가 아니라면 return
        4.2 me와 child가 red라면 balancing 시작
5. 형제노드가 black 또는 NULL인지 red인지 확인
        5.1 형제노드가 black 또는 NULL이면 restruct실행
                5.1.1 me > parent 우측편향
                        5.1.1.1 me > child = RL : child를 parent의 우측에 연결한다 -> child를 기준으로 left rotation 실행
                        5.1.1.2 me < child = RR : left rotation 실행
                        5.1.1.3 me의 parent가 top_root가 아니라면 회전 후 me의 오른쪽 자식 색상을 black으로 변경
                5.1.2 me < parent 좌측편향
                        5.1.2.1 me < child = LR : child를 parent의 좌측에 연결한다 -> child를 기준으로 right rotation 실행
                        5.1.2.2 me > child = LL : right rotation 실행
                        5.1.2.3 me의 parent가 top_root가 아니라면 회전 후 me의 왼쪽 자식 색상을 black으로 변경
                5.1.3 me의 parent가 top_root라면 회전 후 me와 me의 좌우측 색상을 black으로 변경
        5.2 형제노드가 red이면 re-color실행
                5.2.1 parent가 top_root이면 top_root의 좌,우 black으로 변경
                5.2.2 parent가 top_root가 아니면 parent = red, parent 좌우 black
*/
```

RB트리 삽입전략

- RB 트리 삽입 검토
  - 50개의 데이터 삽입 검토 및 프로그램 결과

```
//int data[] = {6, 48, 17, 47, 13, 49, 50, 25, 27, 29, 46, 33, 19, 5, 44, 10, 32, 30, 23, 26, 15, 12, 38, 20, 18, 24, 4, 2,
                //43, 9, 7, 37, 45, 40, 16, 42, 31, 3, 11, 8, 14, 36, 35, 22, 34, 41, 39, 21, 28, 1};
```



```
insert clear!
data =     1     parent =     2     left = NULL     right = NULL     color = red
data =     2     parent =     4     left =     1     right =     3     color = black
data =     3     parent =     2     left = NULL     right = NULL     color = red
data =     4     parent =     6     left =     2     right =     5     color = black
data =     5     parent =     4     left = NULL     right = NULL     color = black
data =     6     parent =    10     left =     4     right =     8     color = black
data =     7     parent =     8     left = NULL     right = NULL     color = black
data =     8     parent =     6     left =     7     right =     9     color = black
data =     9     parent =     8     left = NULL     right = NULL     color = black
data =    10     parent =    27     left =     6     right =    17     color = red
data =    11     parent =    12     left = NULL     right = NULL     color = red
data =    12     parent =    13     left = NULL     right = NULL     color = black
data =    13     parent =    17     left =    12     right =    15     color = black
data =    14     parent =    15     left = NULL     right = NULL     color = red
data =    15     parent =    13     left =    14     right =    16     color = black
data =    16     parent =    15     left = NULL     right = NULL     color = red
data =    17     parent =    10     left =    13     right =    21     color = black
data =    18     parent =    19     left = NULL     right = NULL     color = black
data =    19     parent =    21     left =    18     right =    20     color = black
data =    20     parent =    19     left = NULL     right = NULL     color = black
data =    21     parent =    17     left =    19     right =    23     color = red
data =    22     parent =    23     left = NULL     right = NULL     color = black
data =    23     parent =    21     left =    22     right =    25     color = black
data =    24     parent =    25     left = NULL     right = NULL     color = red
data =    25     parent =    23     left =    24     right =    26     color = black
data =    26     parent =    25     left = NULL     right = NULL     color = red
data =    27     parent =  None     left =    10     right =    38     color = black
data =    28     parent =    29     left = NULL     right = NULL     color = red
data =    29     parent =    30     left =    28     right = NULL     color = black
data =    30     parent =    33     left =    29     right =    32     color = black
data =    31     parent =    32     left = NULL     right = NULL     color = red
data =    32     parent =    30     left =    31     right = NULL     color = black
data =    33     parent =    38     left =    30     right =    36     color = black
data =    34     parent =    35     left = NULL     right = NULL     color = red
data =    35     parent =    36     left =    34     right = NULL     color = black
data =    36     parent =    33     left =    35     right =    37     color = black
data =    37     parent =    36     left = NULL     right = NULL     color = black
data =    38     parent =    27     left =    33     right =    46     color = red
data =    39     parent =    40     left = NULL     right = NULL     color = red
data =    40     parent =    41     left =    39     right = NULL     color = black
data =    41     parent =    43     left =    40     right =    42     color = red
data =    42     parent =    41     left = NULL     right = NULL     color = black
data =    43     parent =    46     left =    41     right =    44     color = black
data =    44     parent =    43     left = NULL     right =    45     color = black
data =    45     parent =    44     left = NULL     right = NULL     color = red
data =    46     parent =    38     left =    43     right =    48     color = black
data =    47     parent =    48     left = NULL     right = NULL     color = black
data =    48     parent =    46     left =    47     right =    49     color = black
data =    49     parent =    48     left = NULL     right =    50     color = black
data =    50     parent =    49     left = NULL     right = NULL     color = red
```

■ RB트리 삭제 구현 전략

EDDI
Electronic Design
Development Institute

- RB 트리 삭제 전략 구성

(1) 삭제하는 노드가 단일 노드 또는 자식이 1개인 노드인지 자식이 2개인 노드인지 확인
(2) 자식이 1개 또는 null인 경우
→ 삭제노드를 그대로 유지한다

(3) 자식이 2개인 노드의 경우
→ 해당 노드 자식의 왼쪽 최대값 또는 오른쪽 최소값을 찾는다
→ 삭제노드를 왼쪽 최대값 노드 또는 오른쪽 최소값 노드로 설정한다

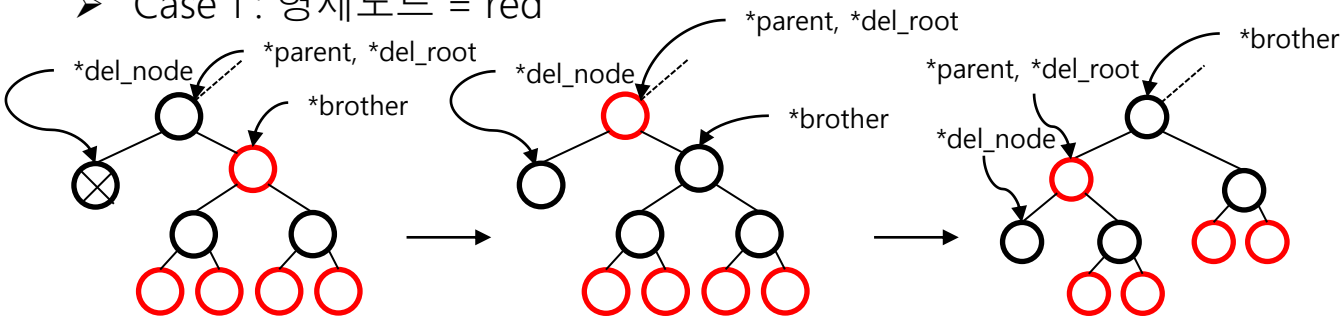
(4) 삭제노드의 색상을 확인

(5) 삭제노드가 red이면 삭제, 밸런싱 미수행

(6) 삭제노드가 black이면
→ 삭제 노드의 자식이 1개이고 red라면 서로 위치를 바꾼 후 삭제
→ 삭제 노드의 자식이 black이거나 null이면 밸런싱 수행

■ RB트리 삭제 구현 전략

- Black노드가 삭제되는 경우
  - ➢ Case 1 : 형제노드 = red



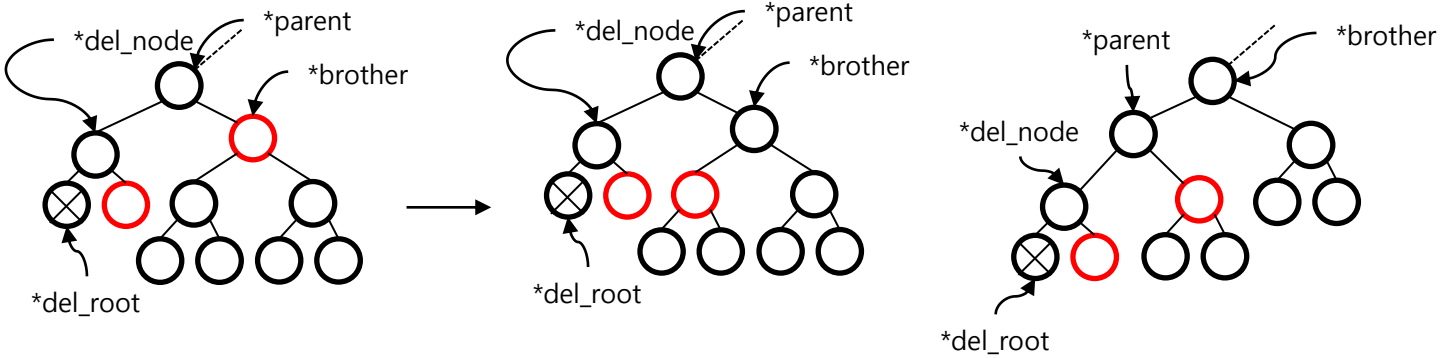
→ 조카의 자식이 red라면 brother의 color와 parent의 color를 서로 바꾼다
→ Brother의 위치가 left/right에 따라 right rotation/left rotation 수행
→ del_root = del_root가 parent와 같다면 회전 후 위치가 변경되므로 right rotation이면 brother의 왼쪽/left rotatio이면 brother의 오른쪽
→ Brother를 기준으로 밸런싱 재귀 수행

▪ RB트리 삭제 구현 전략

EDDI
Electronic Design Development Institute

※ Case 1 예외 상황

➢ Case 2 수행 후 Case 1 수행 시



→ 조카의 자식이 black/null라면 brother의 color와 brother가 right/left이면 brother->left/brother->right의 color를 서로 바꾼다

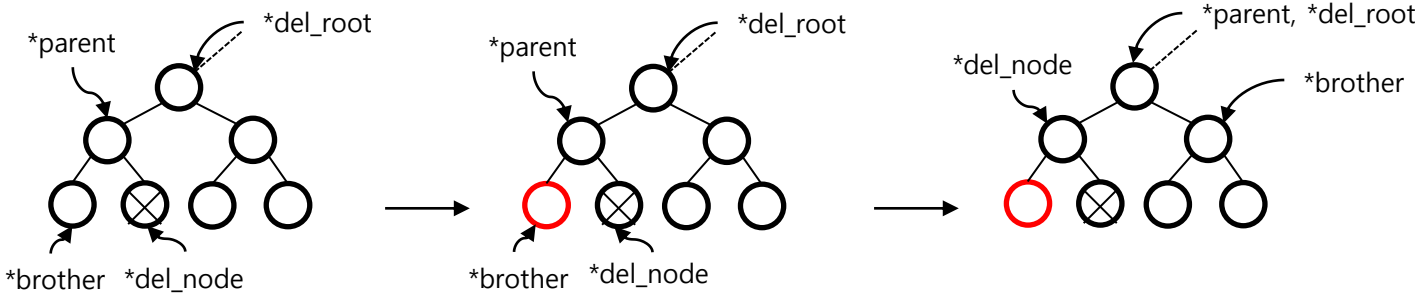
→ Brother의 위치가 left/right에 따라 right rotation/left rotation 수행

→ del_root = del_root가 parent와 같지 않다면 del_root 유지

→ Brother를 기준으로 밸런싱 재귀 수행

■ RB트리 삭제 구현 전략

- Black노드가 삭제되는 경우
  - ➢ Case 2 : 형제노드 = black

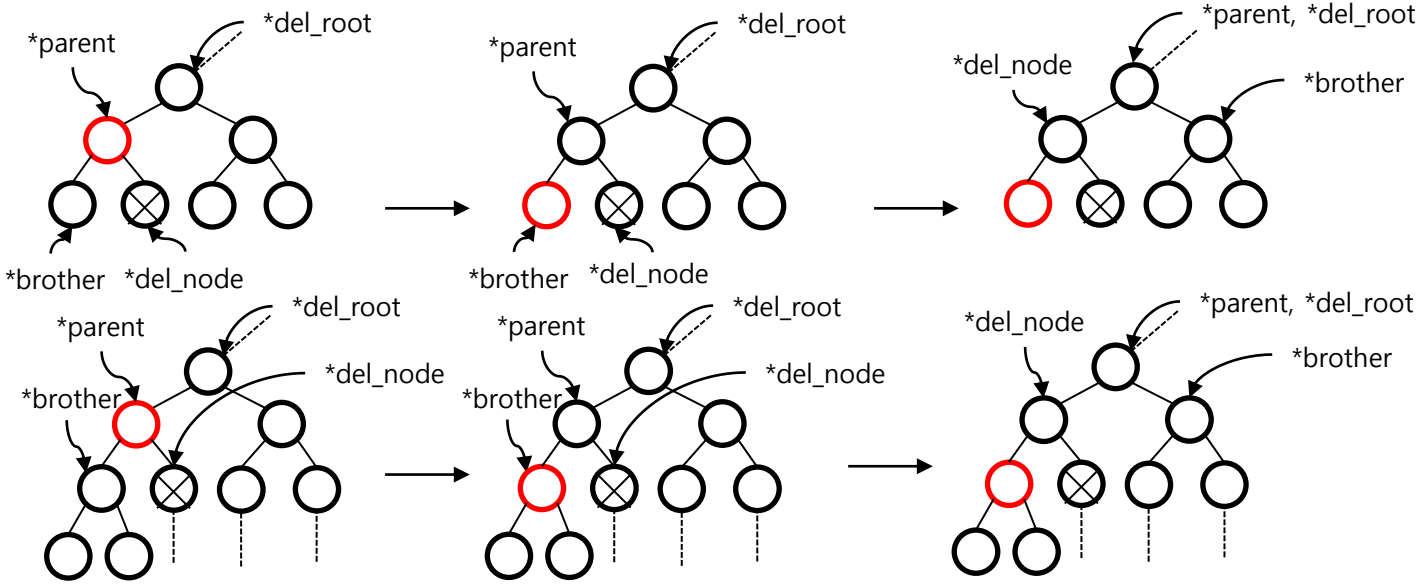


→ Parent가 red가 아니면 brother의 color를 red로 변경한다
→ parent를 기준으로 밸런싱 재귀 수행

■ RB트리 삭제 구현 전략

※ Case 2의 예외 상황

➢ Parent가 red, 조카가 null/black인 경우

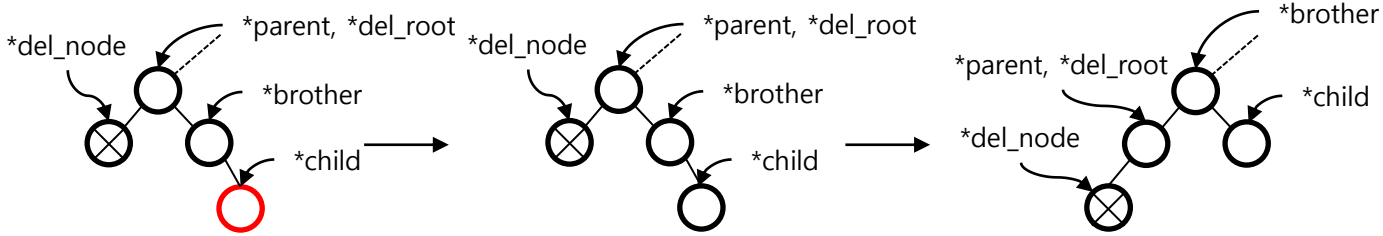


→ Parent가 red이고 조카가 black/null인 경우 parent와 brother의 color를 서로 변경

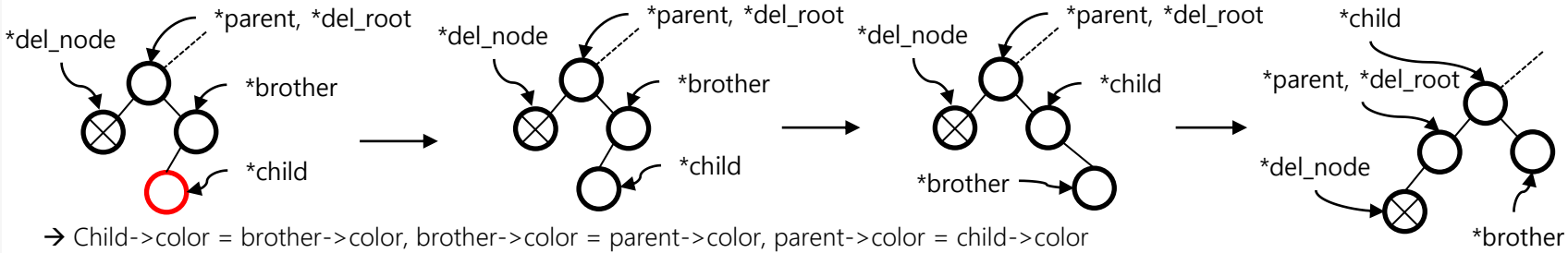
→ return

■ RB트리 삭제 구현 전략

- Black노드가 삭제되는 경우
  - ➢ Case 3(RR, RL) : 형제노드 = black, 조카 = red



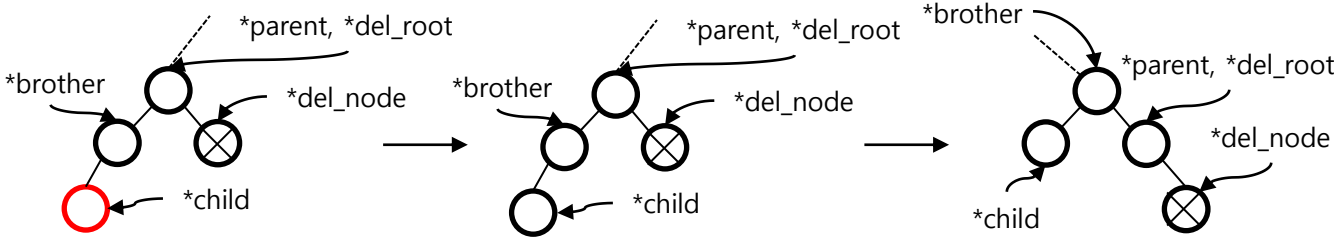
→ Child->color = brother->color, brother->color = parent->color, parent->color = child->color
→ Left rotation 수행 후 del_root가 parent가 같으면 del_root = child->right로 변경한다.



→ Child->color = brother->color, brother->color = parent->color, parent->color = child->color
→ child->right가 있으면 child->right->parent = brother, brother->left = child->right
→ Child->parent = parent, brother->parent = child, parent->right = child
→ Left rotation 수행 후 del_root가 parent가 같으면 del_root = child->right로 변경한다.

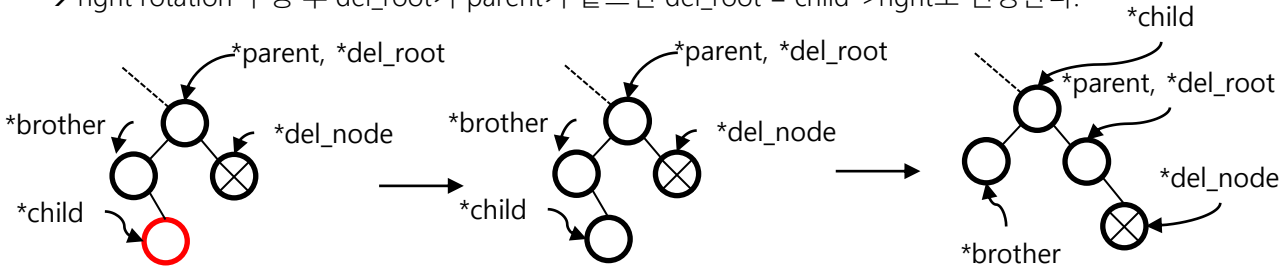
▪ RB트리 삭제 구현 전략

- Black노드가 삭제되는 경우
  - ➢ Case 3(LL, LR) : 형제노드 = black, 조카 = red



→ Child->color = brother->color, brother->color = parent->color, parent->color = child->color

→ right rotation 수행 후 del_root가 parent가 같으면 del_root = child->right로 변경한다.



→ Child->color = brother->color, brother->color = parent->color, parent->color = child->color

→ child->left가 있으면 child->left->parent = brother, brother->right = child->left

→ Child->parent = parent, brother->parent = child, parent->right = child

→ right rotation 수행 del_root가 parent가 같으면 del_root = child->right로 변경한다.