EDDI
Electronic Design
Development Institute

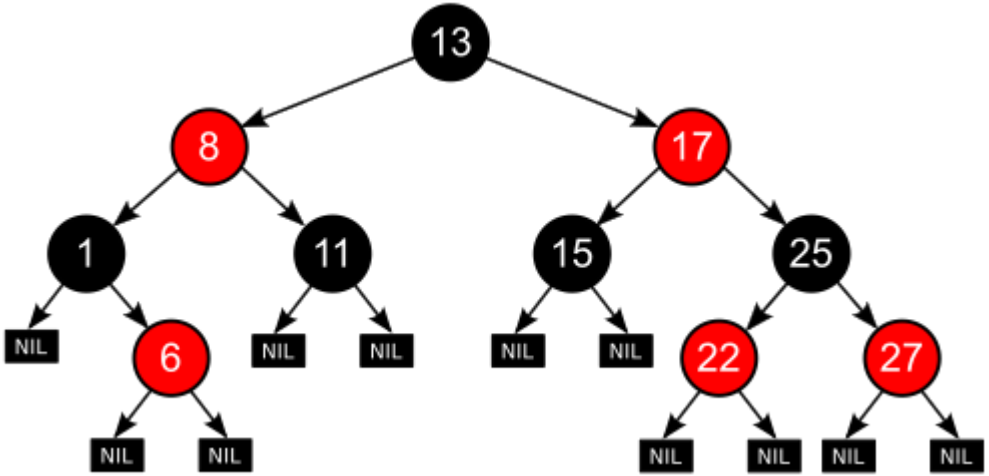
에디로봇아카데미
임베디드 마스터 Lv2 과정

제 1기

2022. 1. 07

김태훈

RED BLACK TREE



AVL Tree가 balance factor를 1 초과가 안되도록 관리하기 때문에 insert, delete때마다 수많은 연산이 발생하게 된다.

이런 문제를 방지하기 위해서
한쪽 tree보다 다른쪽 tree가 2배이상 못 크도록 설계한 tree가 red black tree이다.

현대에 많이 쓰이는 자료구조이다.

RED BLACK TREE

EDDI
Electronic Design Development Institute

Red-Black RULES:

1/ each node must be either **RED** or **BLACK**

2/ the root of the tree must ALWAYS be **BLACK**

3/ two **RED** nodes can never appear in a row within the tree; a **RED** node must always have a **BLACK** parent node, and **BLACK** child nodes

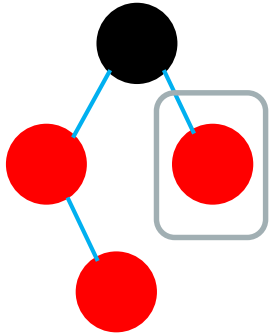
4/ every branch path from the root node in the tree to a NULL pointer passes through the exact same number of **BLACK** nodes (this is also an unsuccessful search path)

RED BLACK TREE

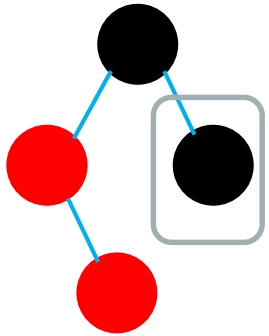
EDDI
Electronic Design
Development Institute

Insert의 경우

RED Violation



삼촌이 RED : Recolor



삼촌이 BLACK : 회전 & Recur

RED BLACK TREE

EDDI
Electronic Design
Development Institute

Delete는?

복잡하다!

RED BLACK TREE

EDDI
Electronic Design Development Institute

Duality in Maxwell Equation

Electric quantities	Magnetic quantities
$\bar{E}_e$	$\bar{H}_m$
$\bar{H}_e$	$-\bar{E}_m$
$\bar{D}_e$	$\bar{B}_m$
$\bar{B}_e$	$-\bar{D}_m$
$\bar{A}$	$\bar{F}$
ϕ_e	ϕ_m
$\bar{J}$	$\bar{M}$
ρ_e	ρ_m
ϵ	μ
μ	ϵ
σ_e	σ_m
PEC	PMC

$$\bar{\nabla} \cdot \bar{D}_e = \rho_e$$
$$\bar{\nabla} \times \bar{E}_e = i\omega \bar{B}_e$$
$$\bar{\nabla} \cdot \bar{B}_e = 0$$
$$\bar{\nabla} \times \bar{H}_e = -i\omega \bar{D}_e + \bar{J}$$

$$\bar{\nabla} \cdot \bar{D}_m = 0$$
$$\bar{\nabla} \times \bar{E}_m = i\omega \bar{B}_m - \bar{M}$$
$$\bar{\nabla} \cdot \bar{B}_m = \rho_m$$
$$\bar{\nabla} \times \bar{H}_m = -i\omega \bar{D}_m$$

RED BLACK TREE

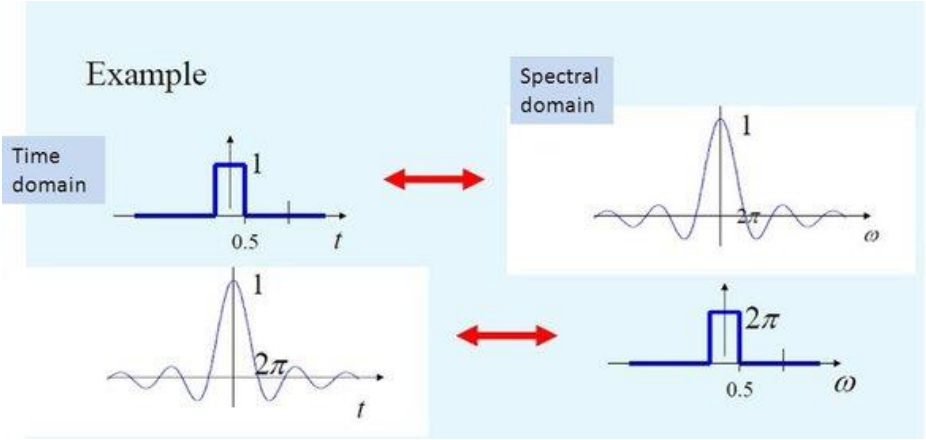
Duality in Fourier Transform

If $x(t) \xleftrightarrow{FT} X(\omega)$ then,

$X(t) \xleftrightarrow{FT} 2\pi \times (-\omega)$

Therefore, using duality property

$$\therefore \frac{1}{1+t} \xleftrightarrow{FT} 2\pi\, e^{\Omega}\, u(-\Omega)$$



RED BLACK TREE

EDDI
Electronic Design Development Institute

Insert와 Delete는 Duality 관계에 있지 않을까?

Insert	Delete
Red Violation	Black Level Violation
See Uncle	See Nephew
Parent – Me가 모두 RED = Recur	Bro, Nephew가 모두 Black = Recur
Rotation 후 Brother가 안 생김	Rotation 후 Brother가 생길 수 있음

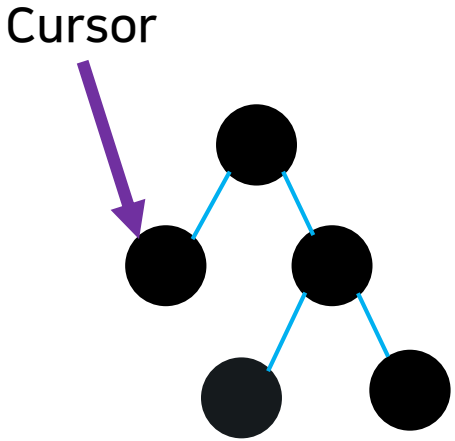
RED BLACK TREE

EDDI
Electronic Design
Development Institute

Delete는 두 가지 때문에 Insert보다 복잡함

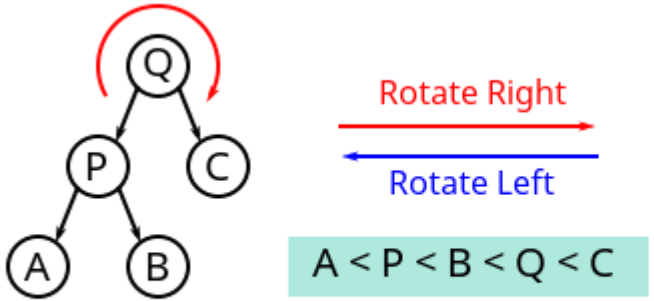
1. Parent, Bro, Nephew가 모두 Black일 때 parent로 올라가서 재조사 해야 함.
2. Rotation 후 Brother가 생겨서 그로 인한 변화를 조사해야 함.

RED BLACK TREE



이 경우는 왼쪽의 Black level이 오른쪽보다 1 작은 경우인데, 회전을 해서는 답이 없으니 형의 색을 Red로 바꾸어서 양쪽 level을 맞추고
Cursor를 Parent로 옮겨서 Recur을 한다.
Recur을 해야 하는 이유는 Parent입장에선 Black level이 -1된 것이기 때문이다.

RED BLACK TREE



회전하면 왼쪽 그림과 같이 P의 자식은 B가 Q의 왼쪽 자식이 되거나, Q의 자식인 B가 P의 오른쪽 자식이 된다.
이 때 문제가 생길 수 있으므로 한번 더 조사한다.

RED BLACK TREE

앞에 두 가지에 주의해서 크게 3가지로 나눈다.

1. 부모가 Black 형제가 Black
2. 부모가 Black 형제가 Red
3. 부모가 Red 형제가 Black

2번 케이스는 형제가 무조건 Black 쌍 자식을 가지므로 조카의 색을 조사할 필요가 없다.

1번 3번 케이스는 조카의 색을 조사해서 각각에 맞는 사항을 해 주어야한다.
보통 Black level을 만족하도록 회전하고 색을 변경해주면 끝난다.

RED BLACK TREE

EDDI
Electronic Design
Development Institute

10000개 Random Sample으로 검증 완료

```
while start
After setting
(*tmp_parent) data : 4726
(*tmp_bro) data : 5684
(*cursor) data : -1
(*cursor)->color == BLACK
(*tmp_parent)->color == RED
BRB
Now root is 7258
delete data : 9692
(*cursor) data : 9692
Now root is 7258
delete data : 4726
(*cursor) data : 4726
((*cursor)->parent) data : 7258
(*cursor) data : 5684
Now root is 7258
delete data : 7258
(*cursor) data : 7258
while start
After setting
(*tmp_parent) data : 5684
(*tmp_bro) data : 8548
(*cursor) data : -1
(*cursor)->color == BLACK
BBB
THERE IS NO NEPHEW
(*tmp_bro) data : 8548
(*tmp_bro)->color = RED:
(*tmp_parent) data : 5684
(*cursor) data : 5684
(*tmp_parent) data : 5684
CONTINUE
Now root is 5684
delete data : 8548
(*cursor) data : 8548
Now root is 5684
delete data : 5684
(*cursor) data : 5684
NIL free
The color of root node is BLACK
```