

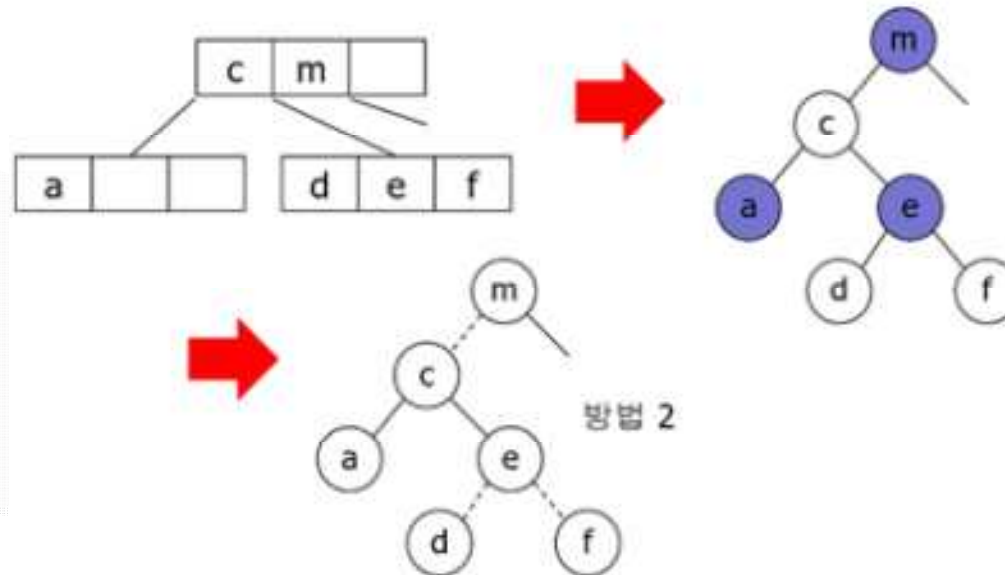


# 레드-블랙 트리

SSAFY 9기 구미 6반 김도연

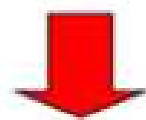
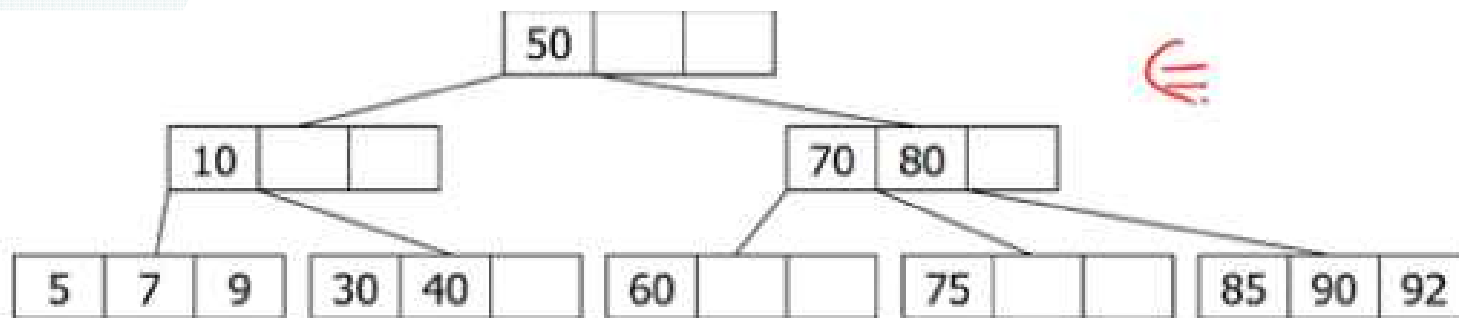
## 2-3-4 트리

- ◆ 최대 키 3개, 최대 자식 4개 인 트리
- ◆ 모든 리프노드가 동일 레벨
- ◆ 2-3-4트리를 이진트리로 표현한것이 레드 블랙 트리
- ◆ 2-3-4를 레드-블랙으로 표시하는 방법은 두가지
  - 1. 노드에 빨강 검정 색을 표시
  - 2. 엣지에 빨강 검정 표시

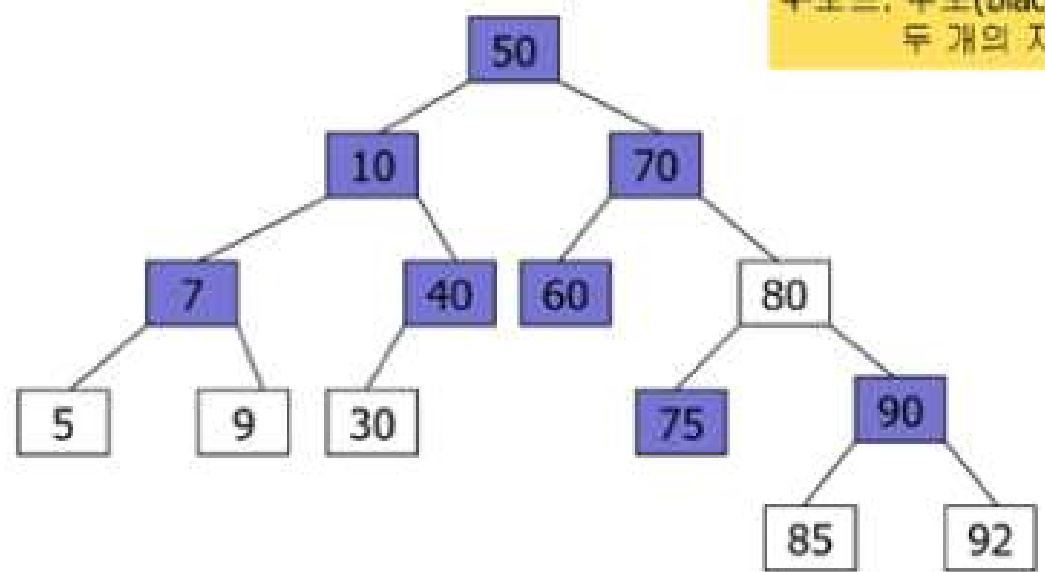


방법 1

방법 2



2-노드: Black node  
 3-노드: 부모(black), 자식(red)  
 4-노드: 부모(black), 두 개의 자식(red)



# 레드-블랙 트리란?

## ◆ 균형 이진 탐색 트리

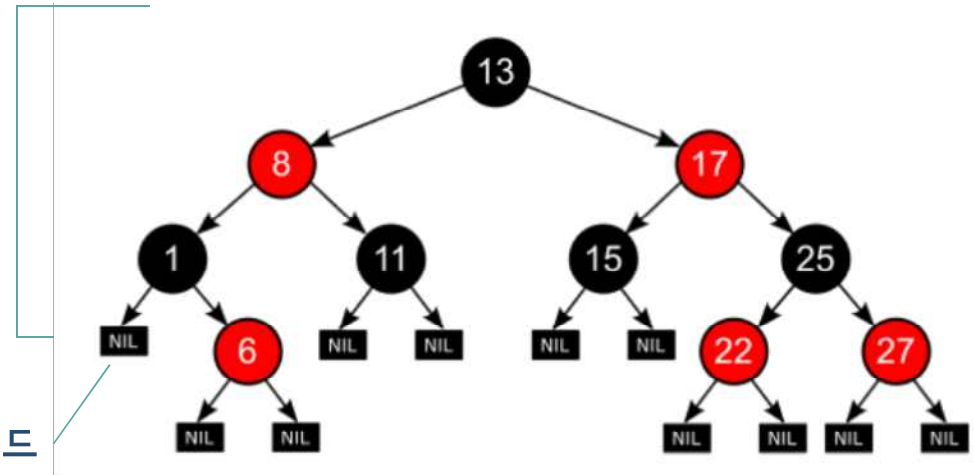
- 균형 이진트리: 좌, 우 트리 높이가 치우치지 않은 트리 (여기서는 블랙 노드 수로 균형이 맞춰진다)

## ◆ 다음의 5개 조건을 만족해야 한다

- 모든 노드는 빨간색 혹은 검은색
- 루트 노드는 검은색이다
- 리프노드는 검은색이다
- 빨간노드는 두개의 자식을 가지며 빨간 노드의 자식은 모두 검정이다 = 빨간 노드가 연속해서 나올수x
- 각노드에서 리프노드 까지 가는 경로가 여러개 있을건데 그 경로들 각각에서 내려가면서 마주치는 검정색 노드의 수가 동일해야 함 = 모든 리프노드에서 Black Depth 동일

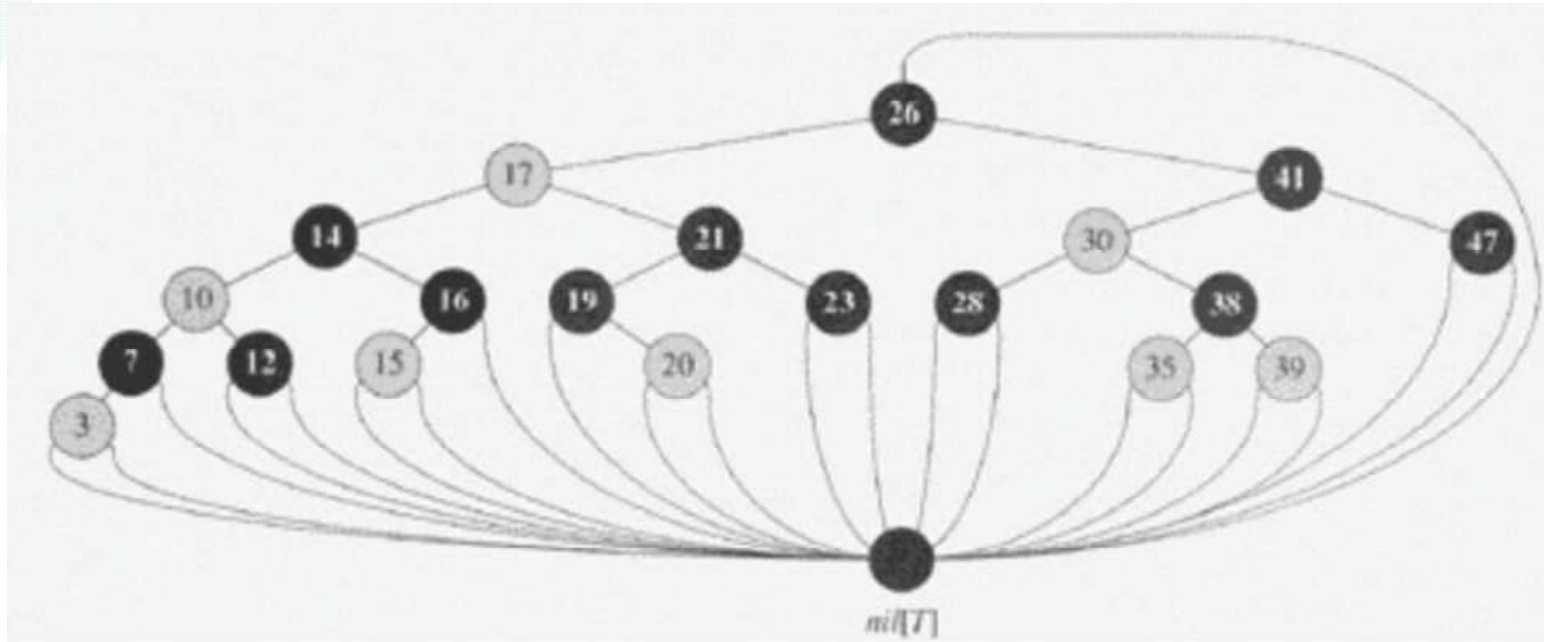
내부노드

리프노드



# 레드 블랙 트리란

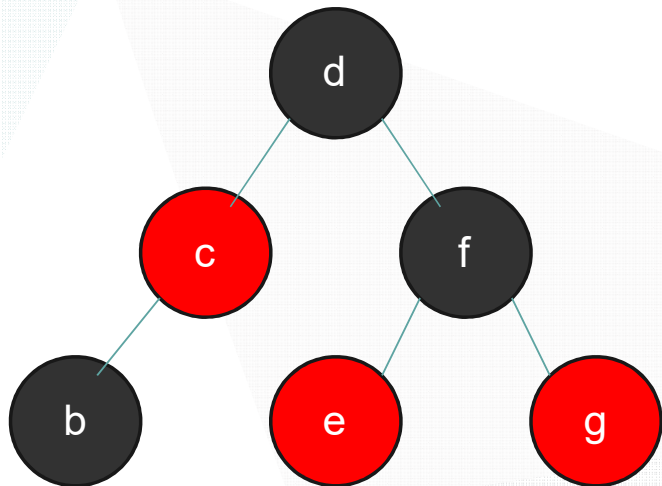
- ◆ Search, Insert, Delete 연산을 최악의 경우에도  $O(\log n)$  시간에 지원
- ◆ 실제 구현할때는 NIL 을 하나하나 다는것이 아니라 NIL을 하나만 만들고 연결



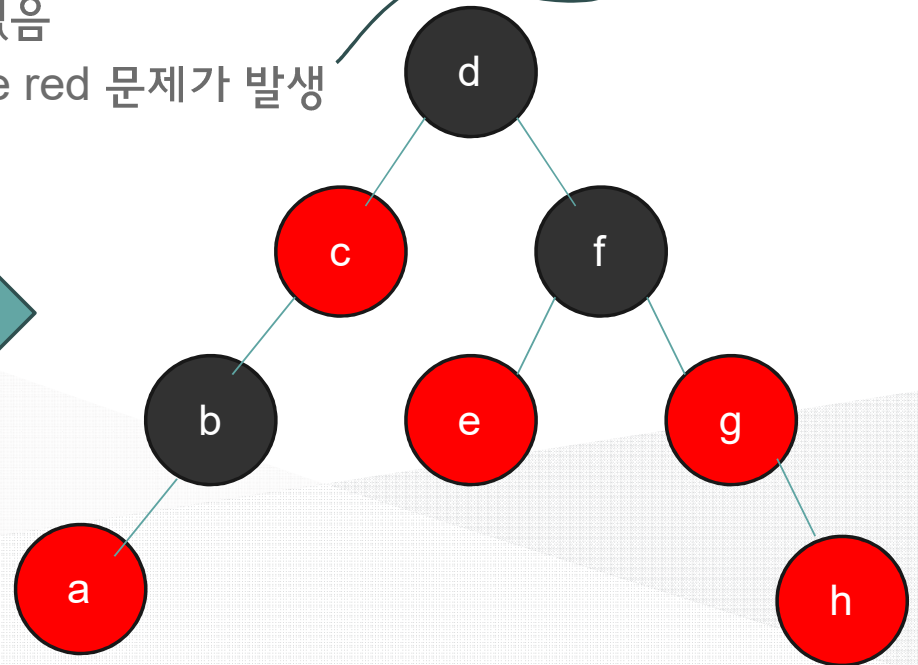
# 레드-블랙 트리: 삽입

삽입을 블랙노드로 안하는 이유: 모든 경로에서 마주치는 블랙 노드 수가 같다는 규칙이 어긋남

- ◆ 삽입은 리프 노드에서 이루어짐
- ◆ 1. 최초로 추가되는 경우, 즉 루트인 경우는 무조건 블랙노드로 추가한다
- ◆ 2. 루트가 아닌 경우에 새로 추가되는 노드는 무조건 레드노드로 추가한다
  - 새로 추가되는 노드의 부모가 블랙인 경우 -> 문제 없음
  - 새로 추가되는 노드의 부모가 레드인 경우 -> double red 문제가 발생



a, h 추가

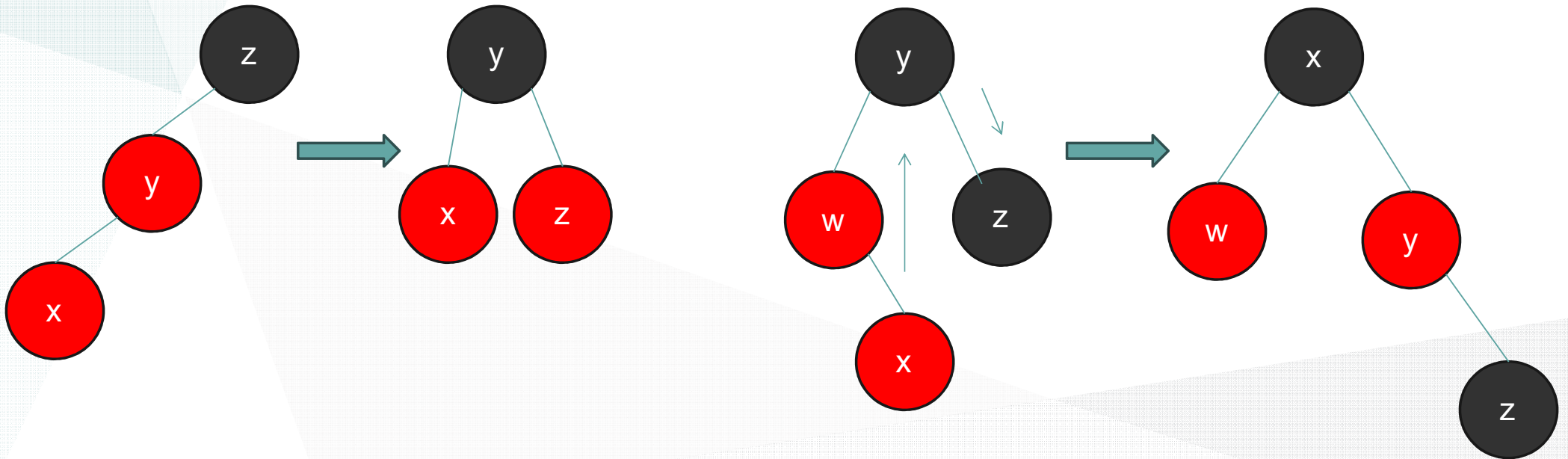


a추가에서는 문제 없음

h 추가에서 double red문제

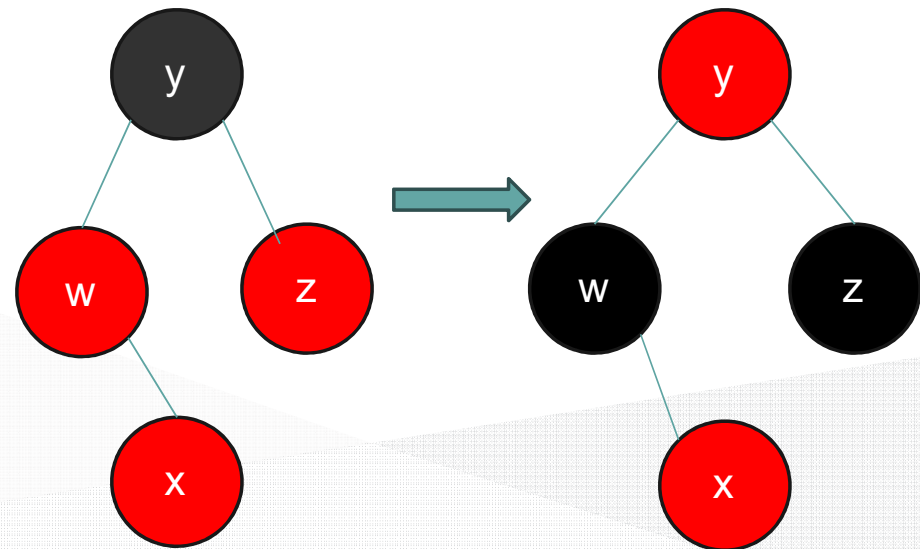
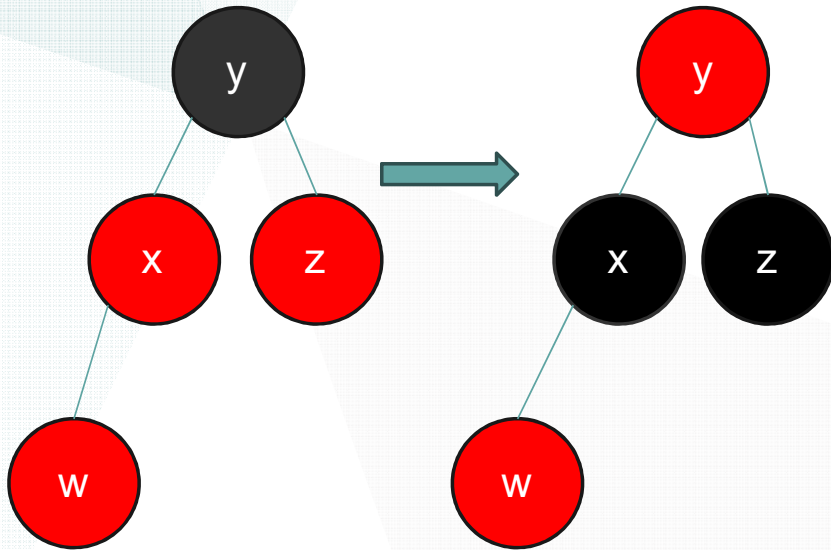
# 레드-블랙 트리: Double Red해결 -1

◆ 새로 추가된 노드의 부모의 형제 노드가 없거나 블랙인 경우 -> 회전



## 레드-블랙 트리: Double Red 해결 -2

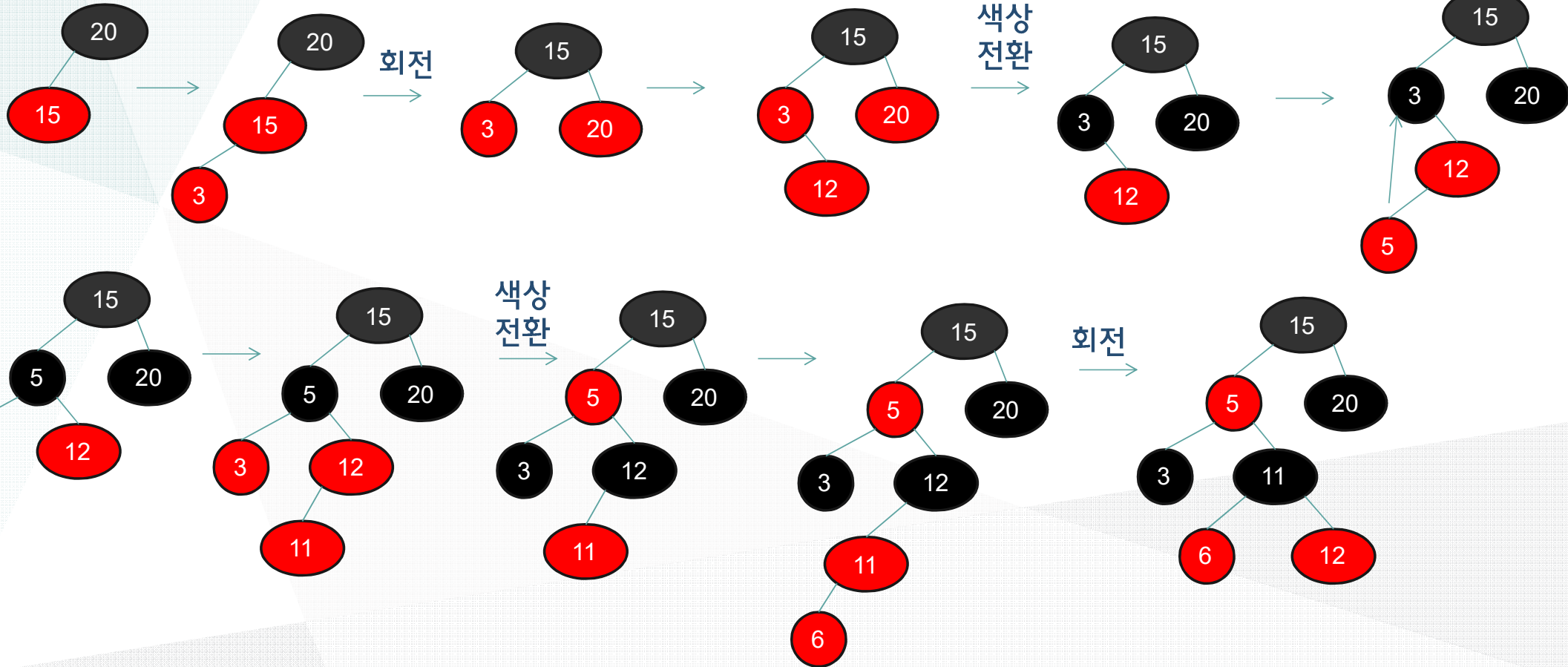
- ◆ 새로 추가된 노드의 부모의 형제 노드가 레드인 경우 -> 추가 노드의 부모의 부모는 빨간색으로, 추가 노드의 부모와 부모의 형제를 블랙으로
- ◆ Bottom up으로 색깔 변경, 마지막에 루트의 색을 블랙으로 변경





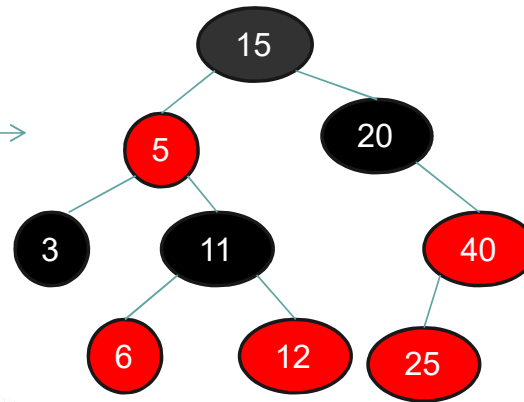
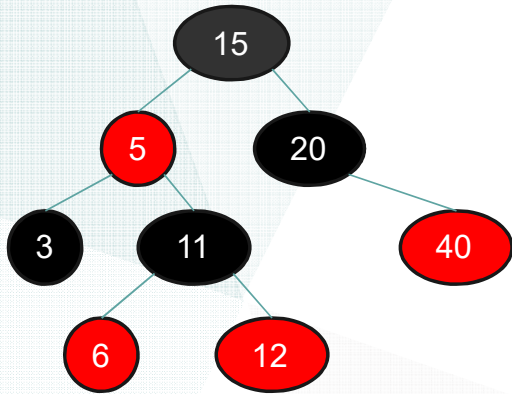
# 삽입 예시

20 15 3 12 5 11 6 40 25 18 순서로 삽입

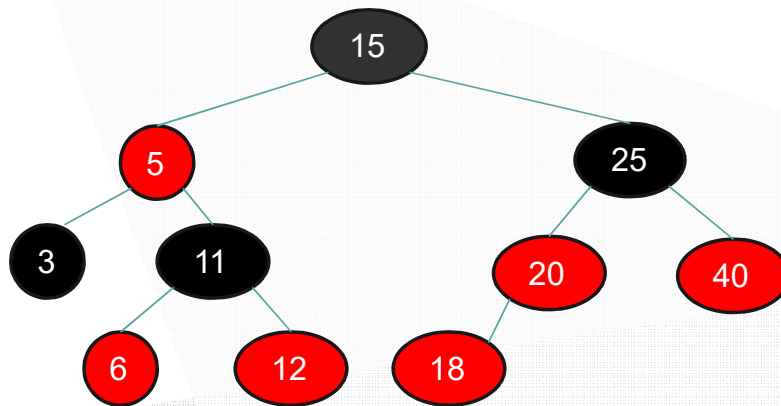
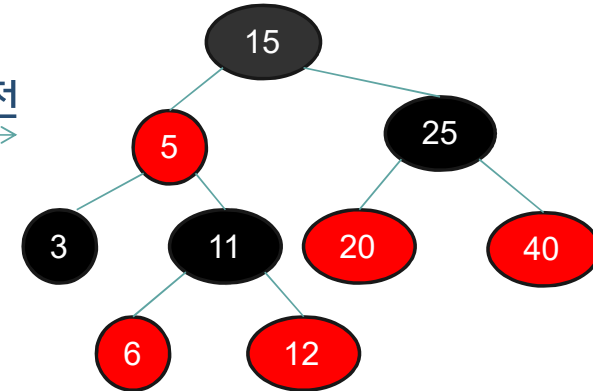


## 삽입 예시 -2

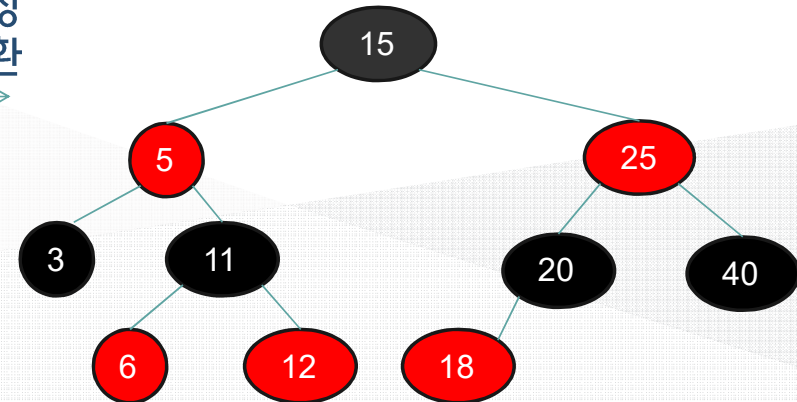
20 15 3 12 5 11 6 40 25 18 순서로 삽입



회전  
→



색상  
전환  
→

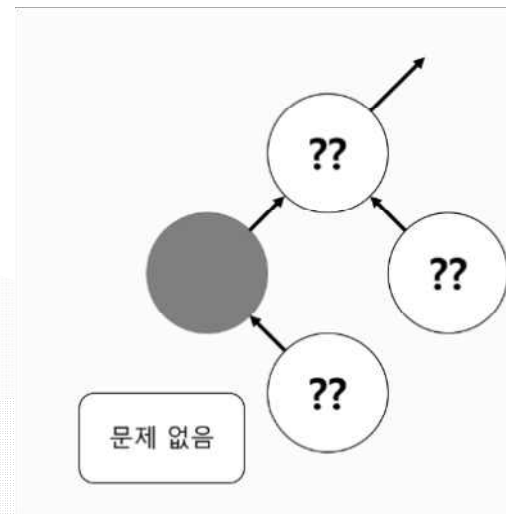
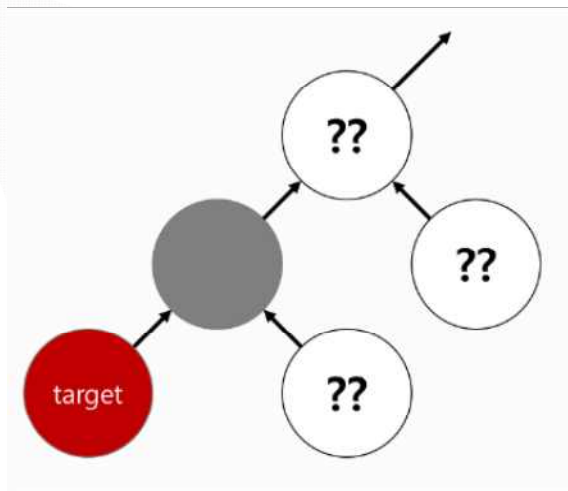


# 레드-블랙 트리: 삭제

## ◆ 삭제 노드가 레드인 경우

이 경우 아무 문제가 없음

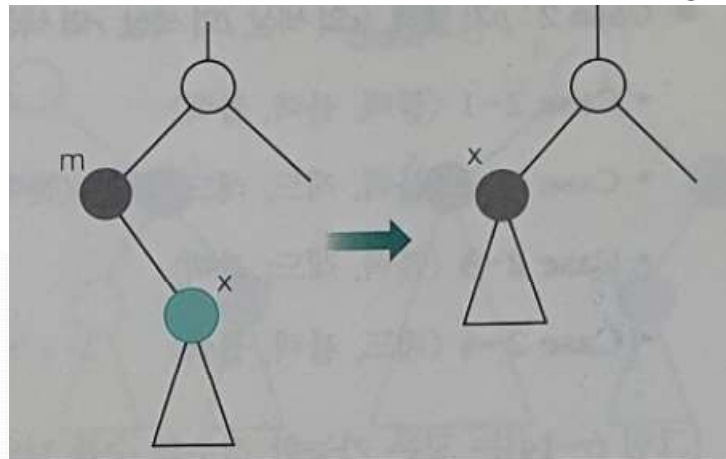
삭제 될 노드가 빨강이라는것은 그 삭제 대상 노드의 부모는 검정  
그냥 삭제 되어도 균형도 맞고 black의 수도 변함 없음



# 레드-블랙 트리: 삭제

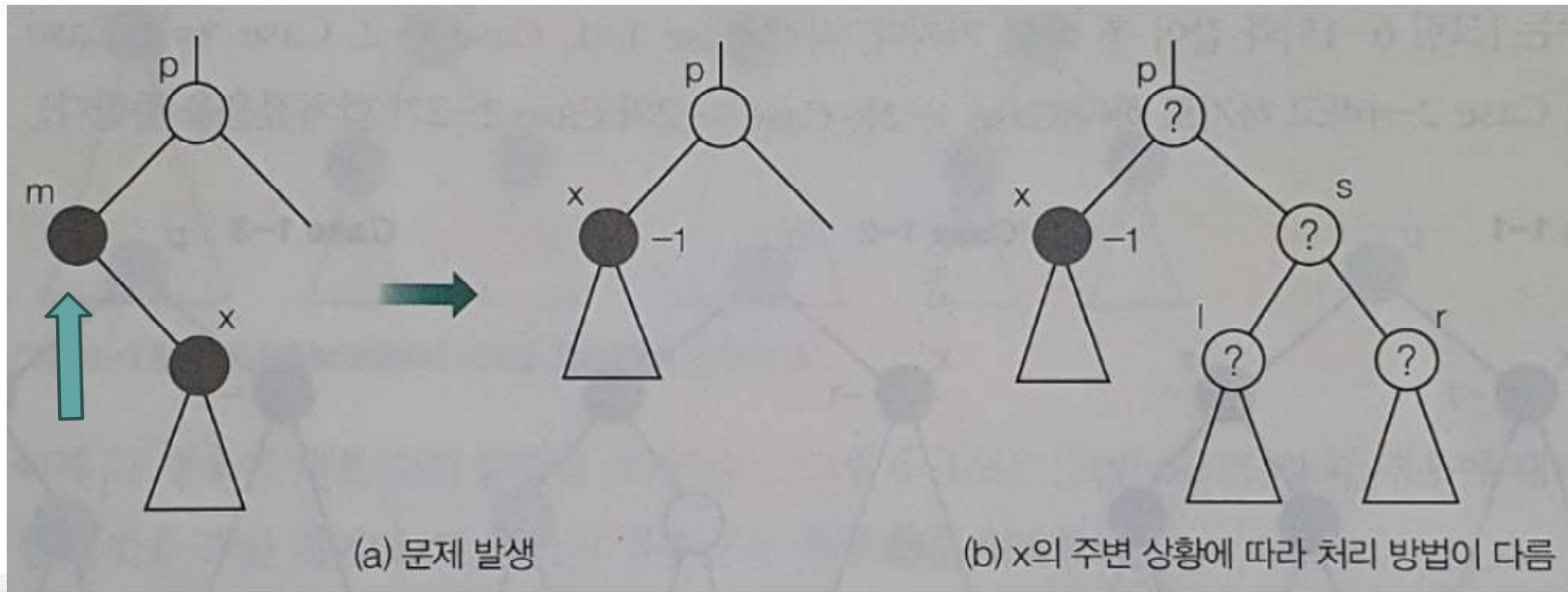
## ◆ 삭제 노드가 블랙

삭제 노드가 블랙이어도 자식이 하나고 레드이면 문제 없다  
m을 삭제하고 자식인 x를 위로 올리고 블랙으로 변경



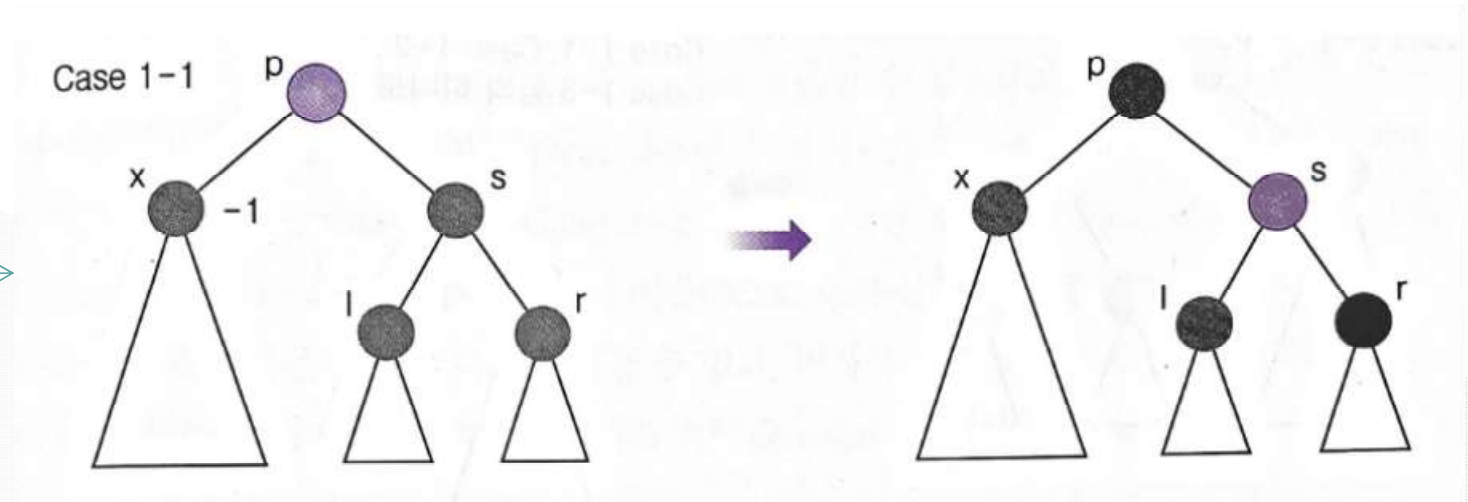
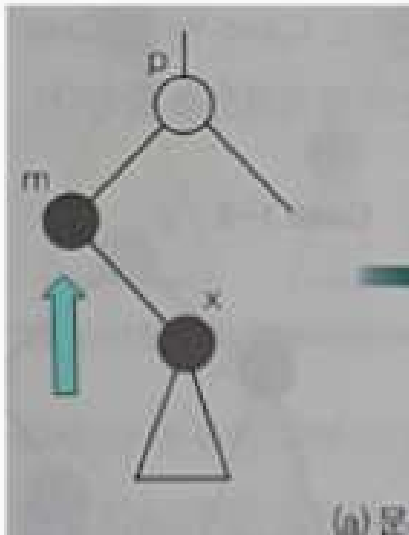
# 레드-블랙 트리: 삭제

- ◆ 삭제 노드가 블랙
- ◆ 삭제노드의 자식도 블랙이거나 자식이 없다면 문제 발생  
블랙 노드를 삭제하면 해당 방향 블랙 노드수가 1감소한다  
그러면 루트에서 리프노드까지 내려가는데 만나는 블랙 노드 수가 달라지게됨



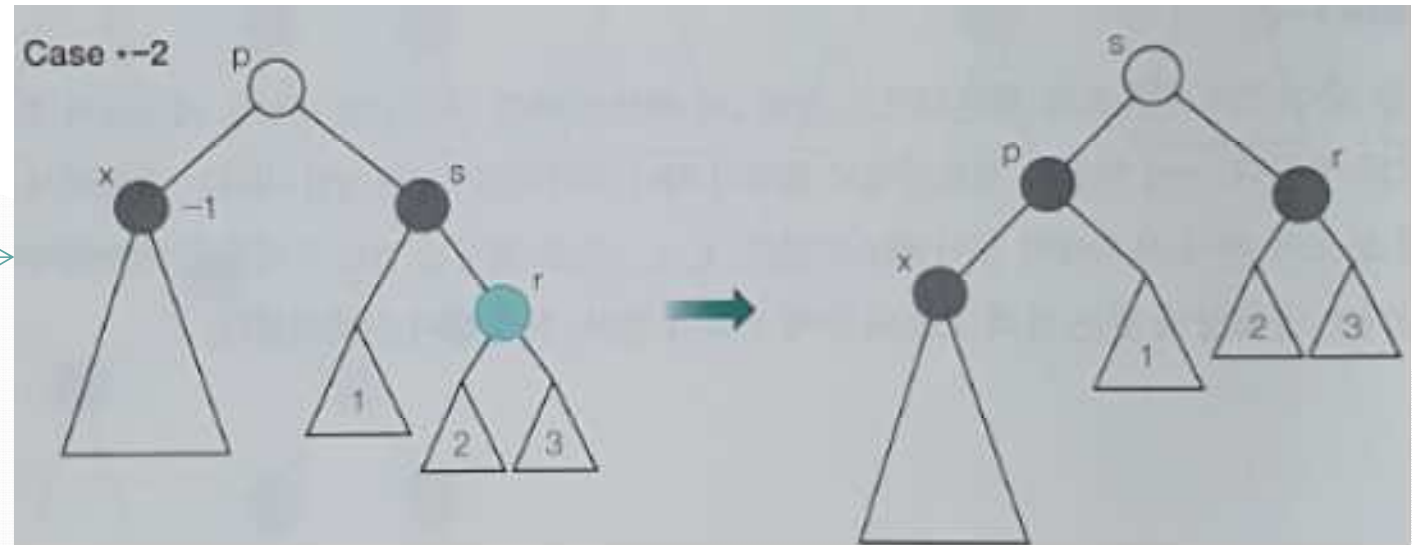
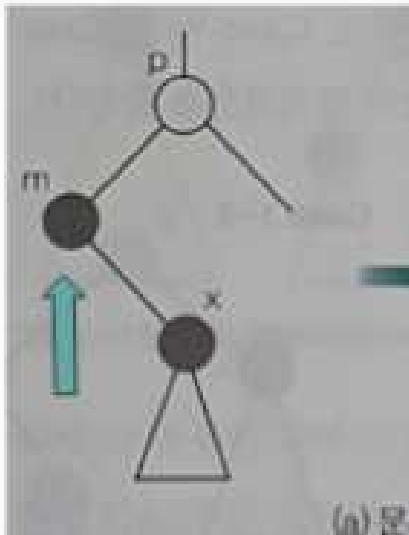
# 레드 블랙 트리 삭제

- ◆ 경우1: 삭제 노드가 블랙, 삭제 노드 부모 레드, 삭제 노드의 자식 블랙, 삭제 노드의 형제가 블랙, 삭제 노드의 형제의 자식모두 블랙
- ◆ -> p와 s의 색상을 바꾼다



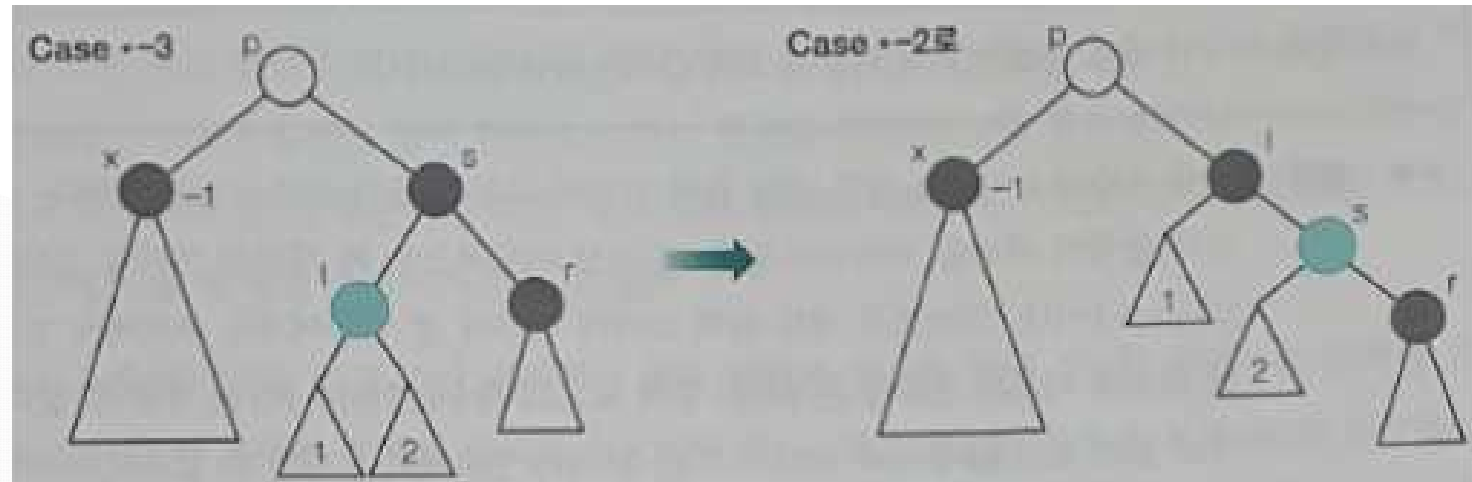
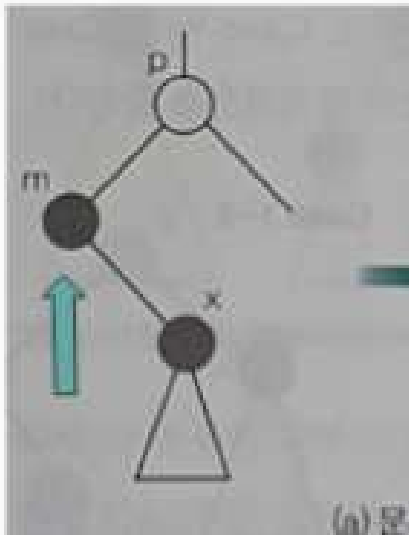
# 레드 블랙 트리 삭제

- ◆ 경우2: 삭제 노드가 블랙, 삭제노드 부모의 색상 무관, 삭제 노드의 자식 블랙, 삭제 노드의 형제가 블랙, 삭제노드의 형제의 자식중 레드가 있는 경우
- ◆ -> p중심으로 회전, p와s의 색상 교환, r의 색상을 블랙으로, 회전, 자식 옮김



# 레드 블랙 트리 삭제

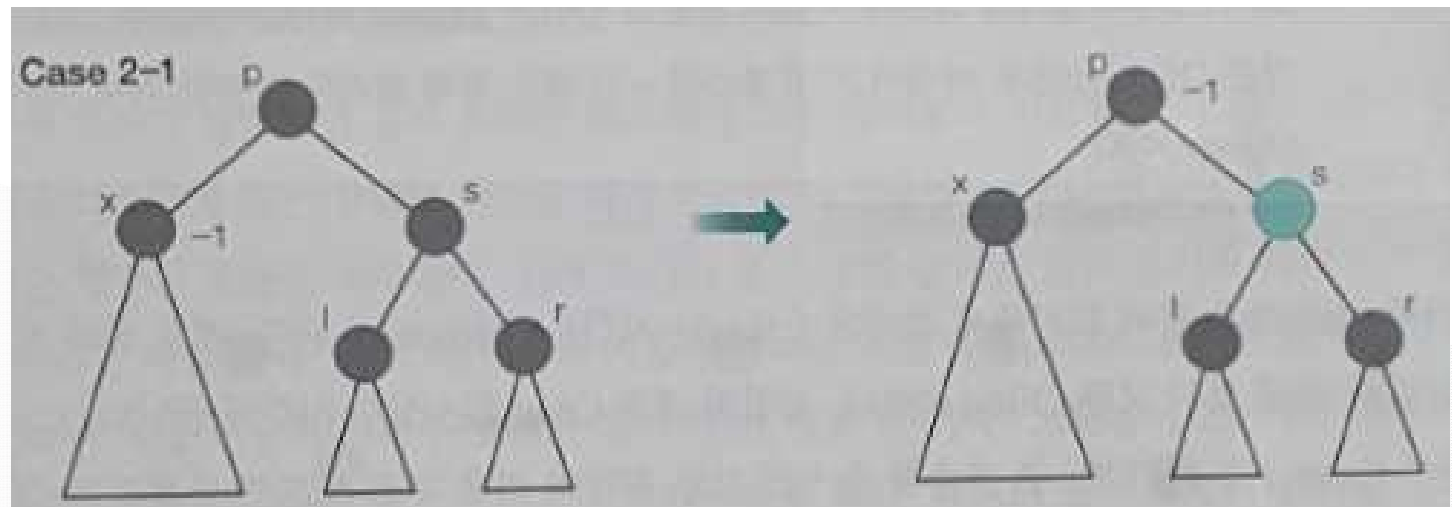
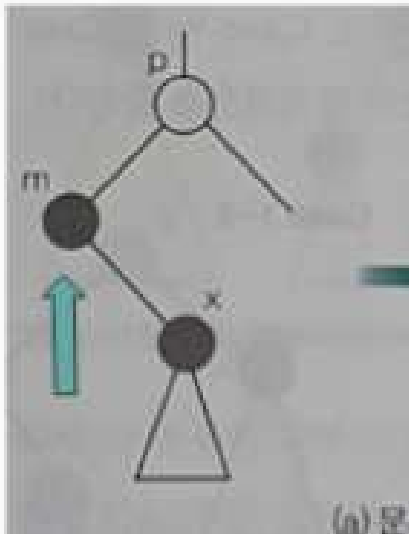
- ◆ 경우3: 삭제 노드가 블랙, 삭제노드 부모 색상 무관, 삭제 노드의 자식 블랙, 삭제 노드의 형제가 블랙, 삭제 노드의 형제의 자식이 레드하나 블랙하나
- ◆ -> s 중심으로 돌리고 l과 s 색상 교환-> 경우2처럼 회전





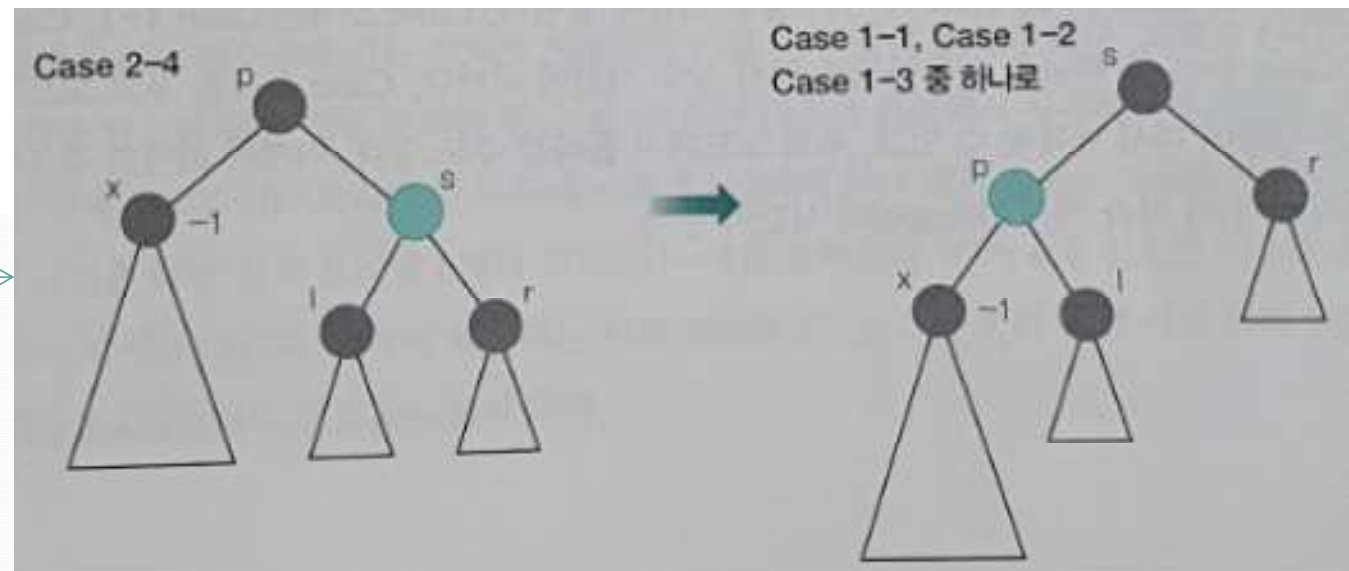
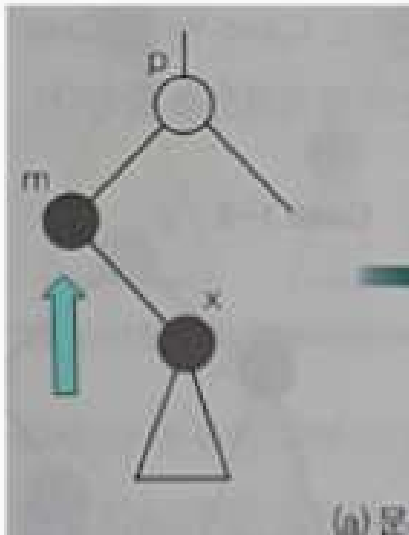
# 레드 블랙 트리 삭제

- ◆ 경우4: 삭제 노드가 블랙, 삭제노드 부모 블랙, 삭제 노드의 자식 블랙, 삭제 노드의 형제가 블랙, 삭제 노드의 형제의 자식이 모두 블랙
- ◆ -> s를 블랙에서 레드로 바꿈, p에서 재귀적으로 처리

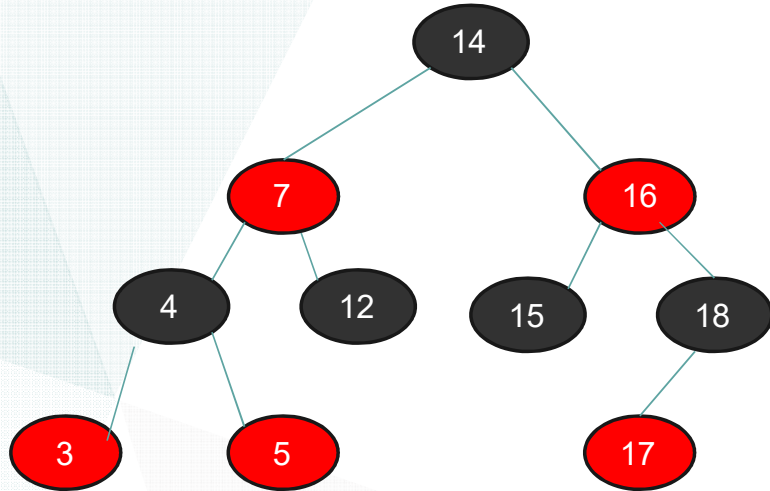


# 레드 블랙 트리 삭제

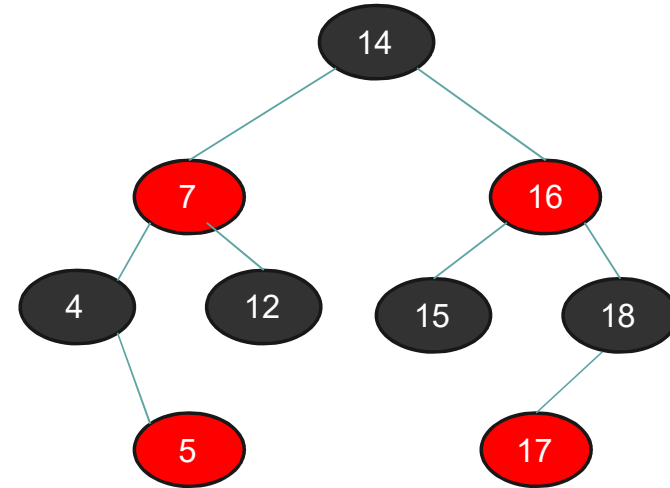
- ◆ 경우5: 삭제 노드가 블랙, 삭제노드 부모 블랙, 삭제 노드의 형제가 레드
- ◆ -> p중심으로 회전, p와 s 색상교환-> 삭제 노드의 부모가 레드였던 경우1과 같아짐



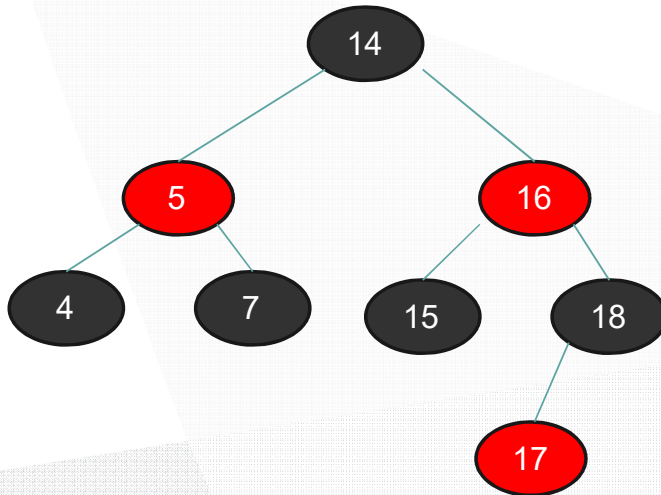
# 레드 블랙 트리 삭제 예시



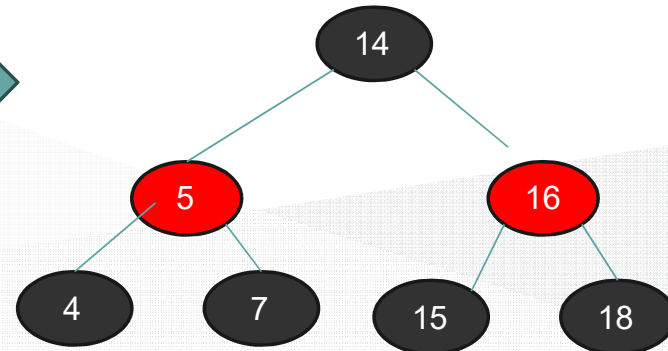
3제거



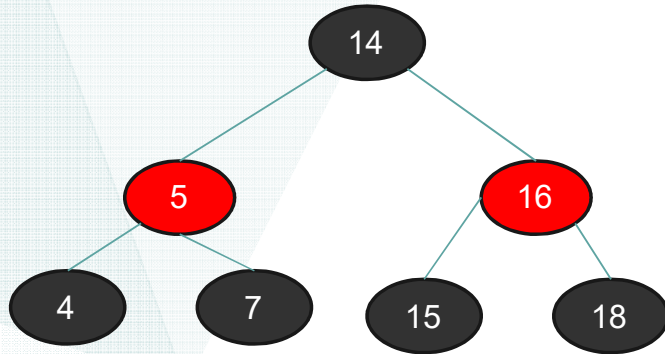
12제거  
경우2 적용



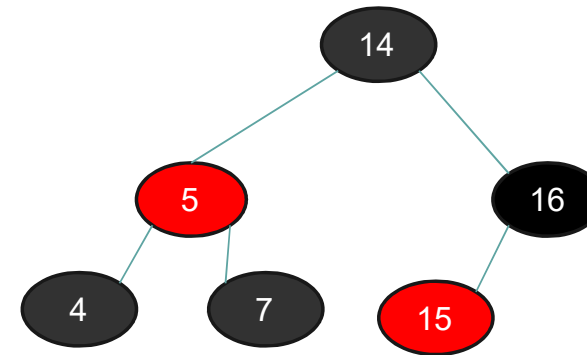
17제거



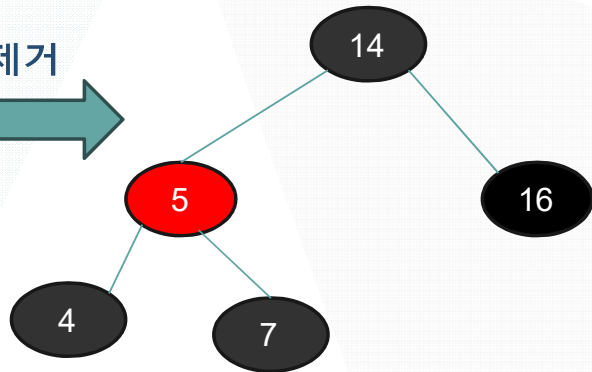
## 레드 블랙 트리 삭제 예시 -2



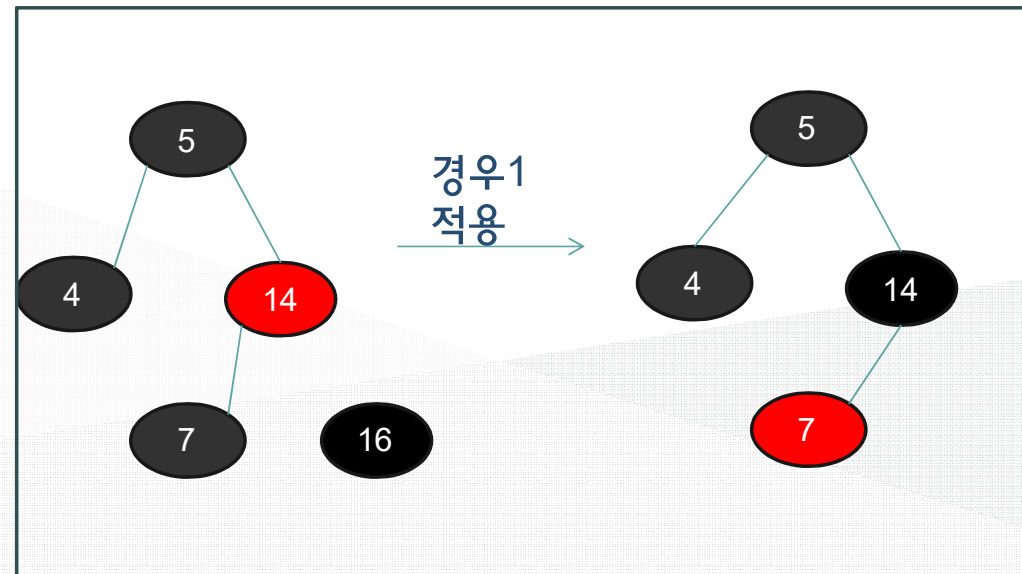
18제거  
경우1 적용



15제거



16제거  
경우5 적용



# 레드블랙 트리 시뮬레이션 사이트

◆ <https://www.cs.usfca.edu/~galles/visualization/RedBlack.html>

# 참고자료

- ◆ 영남대학교 알고리즘 수업 강의자료
- ◆ <https://codetime.tistory.com/444>
- ◆ <https://zeddios.tistory.com/237>
- ◆ <https://dev-game-standalone.tistory.com/94>
- ◆ <https://dev-game-standalone.tistory.com/93>
- ◆ <https://m.blog.naver.com/min-program/221231697752>
- ◆ <https://lemonlemon.tistory.com/135>
- ◆ <https://dad-rock.tistory.com/355>