

규칙

1. 모든 노드는 빨강 or 검정
 2. Root = Black
 3. Leaf = Black
 4. 빨강의 자식 = Black, 검정의 자식 = 상관 X
 5. Root → Leaf 까지 가는 모든 경로에서 만나는 검정노드의 개수는 같다
- * Leaf Node는 NIL로 채운다.
(Sentinel node)

Rotate

- 부모 위치와 자식의 위치를 바꾸는 것

Right - Rotate ⇒ 왼쪽자식과 부모 위치 교환

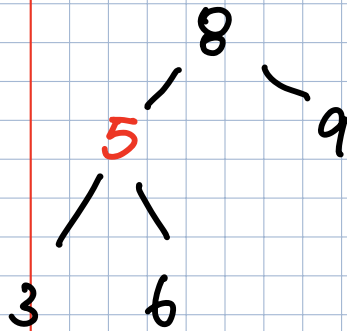
Left - Rotate ⇒ 오른쪽자식과 부모 위치 교환

Title: _____

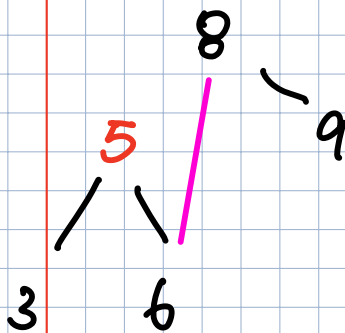
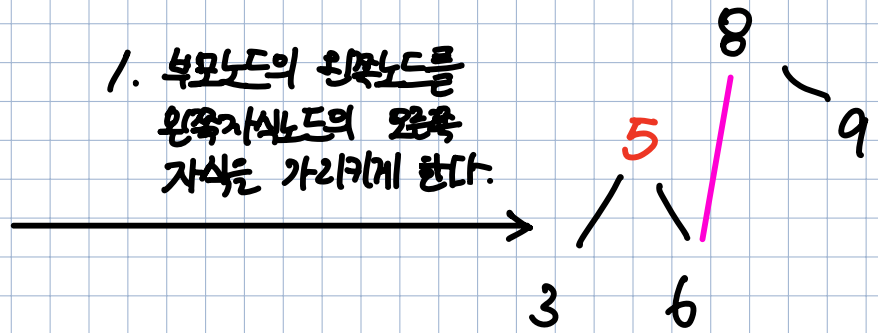
Name: _____

Date: _____

Right Rotate



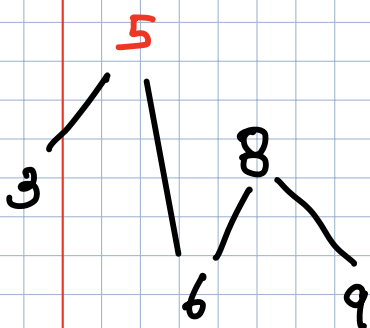
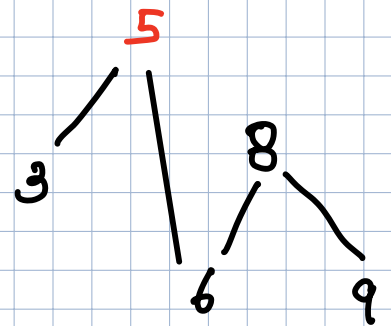
1. 부모노드의 왼쪽노드를
왼쪽자식노드의 왼쪽
자식으로 가리키게 한다.



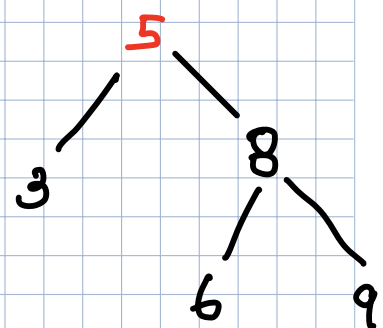
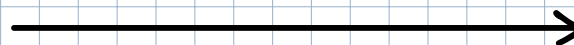
2. Leftchild가
Parent의 부모가 되게
한다.

* 이때 Parent의 부모가
존재하지 않으면
Root가 된다.

* Parent의 부모가 존재한다면
Parent의 부모의 LeftNode로
설정해준다.



3. LeftChild의
RightChild를
Parent로 바꾼다

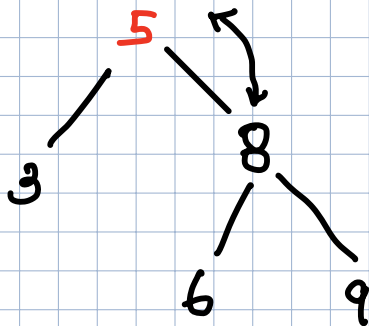


Title: _____

Name: _____

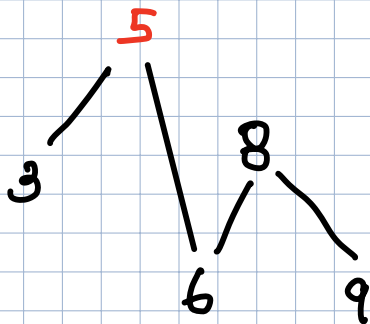
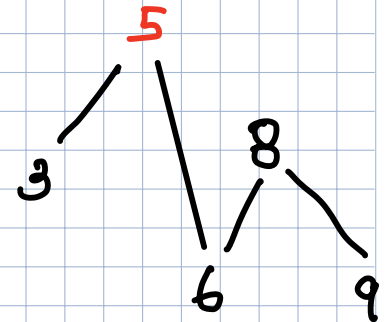
Date: _____

Left Rotate

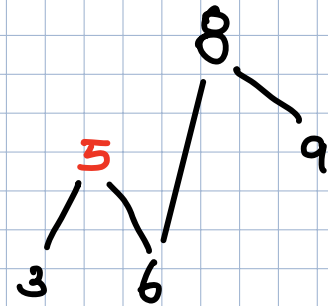


1. 부모노드의 오른쪽 노드를
오른쪽 자식의 왼쪽 자식으로
설정한다.

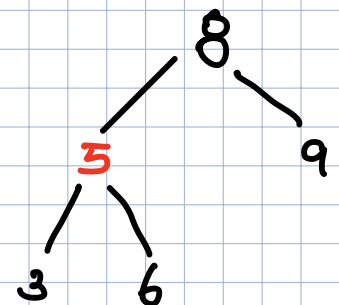
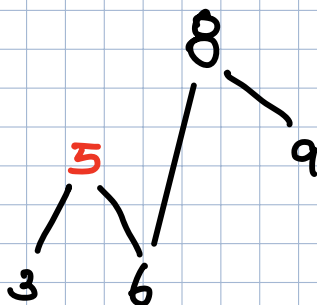
8: RightChild
5: Parent



2. RightChild가 Parent의
Parent가 되게 한다.



3. RightChild의
LeftChild를 Parent로
설정한다.



Title:

Name:

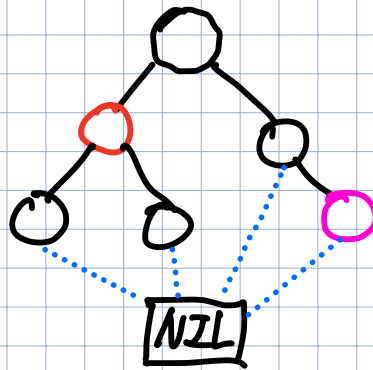
Date:

Insert

새 노드 삽입 → Red 노드로 넣고 자식으로 NIL을 갖게한다

- ① 이진탐색트리에 넣을것처럼 일단 넣는다
- ② 해당노드를 Red로 바꾸고, 자식으로 NIL을 달아준다.
(Black)
- ③ Tree를 Rebuild 한다.

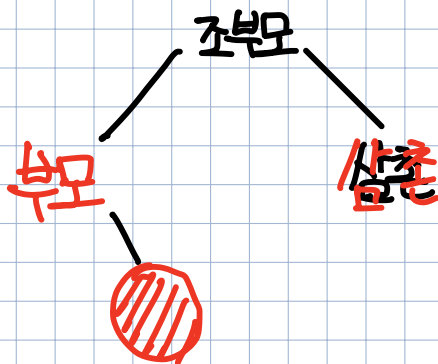
* 실제 모든 LeafNode가 갖는 NIL은 모두 하나의 노드를 가리킨다



노드 삽입시 확인해야 하는것

- ① Root노드가 점성인가? → Root 노드를 무조건 점성으로!!
- ② Red노드의 자식들은 모두 Black인가?

2번 조건을 위반한 경우



- ① 삼촌도 Red 인 경우
- ② 삼촌 = Black, 사촌삼촌한 노드가 부모의 오른쪽 자식인 경우
- ③ 삼촌 = Black, 사촌삼촌한 노드가 부모의 왼쪽 자식인 경우

Title: _____

Name: _____

Date: _____

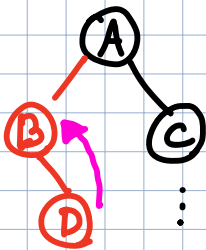
① 삼촌이 Red인 경우

→ 부모와 삼촌을 Black으로, 조부모를 Red로

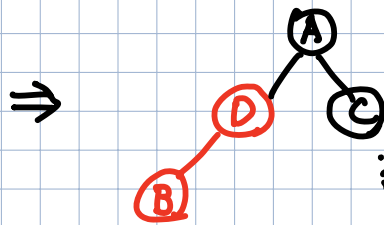
이때 조부모의 부모가 Red라면?

⇒ ①, ②, ③ 다시 체크

② 삼촌 = Black, 삽입된 노드가 부모의 오른쪽 자식인 경우



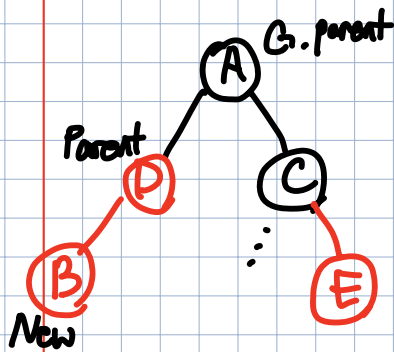
B를 기준으로 D를 Left Rotate



⇒ ③의 경우로 처리한다

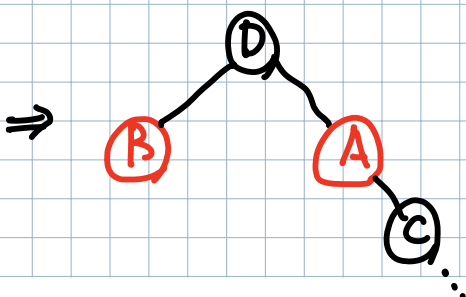
(이제 B를 새로 삽입한 노드로 간주)

③ 삼촌 = Black, 삽입된 노드가 부모의 왼쪽 자식인 경우



1. 할아버지를 Red, 부모를 Black으로

2. 할아버지 노드를 기준으로 Parent를 Right Rotate



Title :

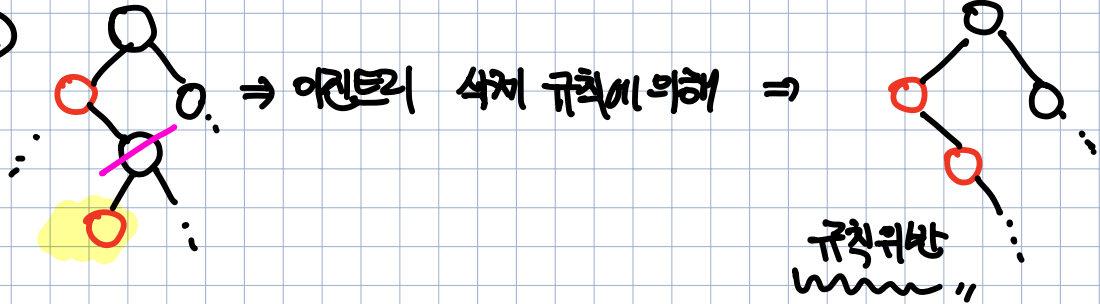
Name :

Date :

Remove

빨간노드 삭제 \Rightarrow 문제 X, 이진트리 삭제방법대로 삭제한다.
검정노드 삭제

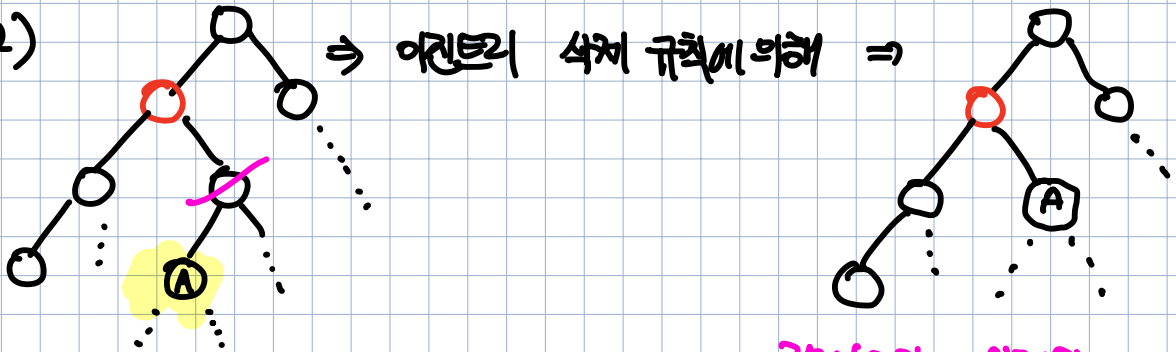
Case 1)



(자식노드는 모두 Black,
Root \rightarrow Leaf의 Black 개수 다름)

\Rightarrow 대체되는 노드를 Black으로 칠한다.

Case 2)



Root \rightarrow Leaf Black 개수가 달라짐

\Rightarrow /번조건인 "모든 노드는 Red/Black이어야 한다"를 위반한 것으로 생각

즉, (A)를 이중 Black color로 생각

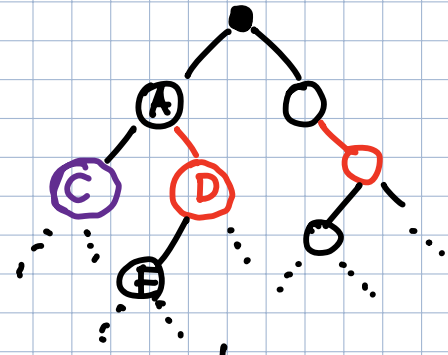
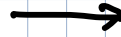
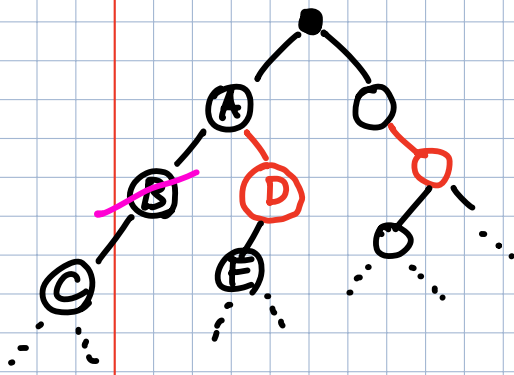
해결방법

1. 형제가 빨간색인 경우
2. 형제가 검정색인 경우
 - A. 형제의 양쪽 자식이 Black인 경우
 - B. 형제의 왼쪽 = Red, 오른쪽 = Black인 경우
 - C. 형제의 오른쪽 = Red인 경우

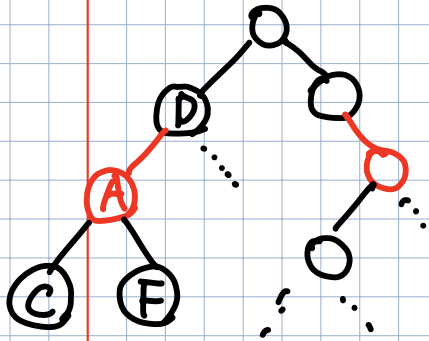
Title: 형제가 삽입된 경우

Name: _____

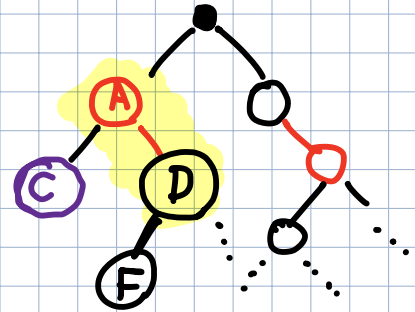
Date: _____



1. 형제를 Black으로 바꾸고 Red로



2. 부모가 오른쪽
Left Rotate



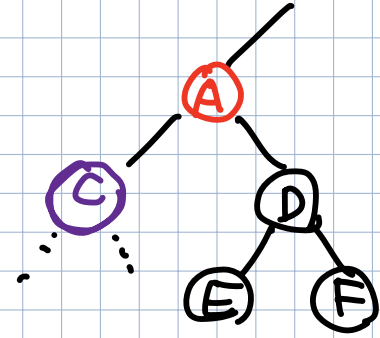
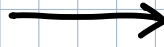
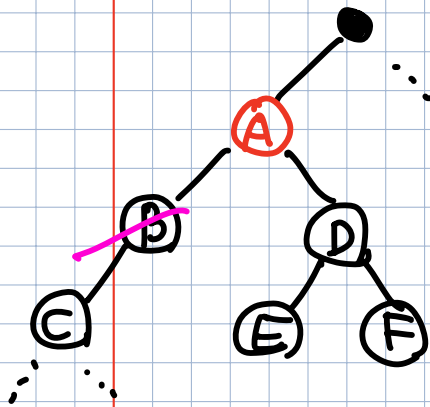
A, B, C의 경우로 처리

Title: 형제가 검정색인 경우

A. 형제와 양쪽 자식이 Black인 경우

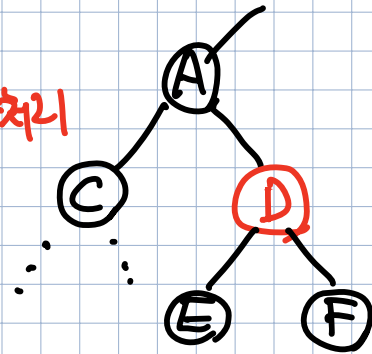
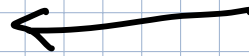
Name: _____

Date: _____



1. 형제를 Red로
부모를 Black으로

2. 부모에게 다시
A, B, C를 따라치기

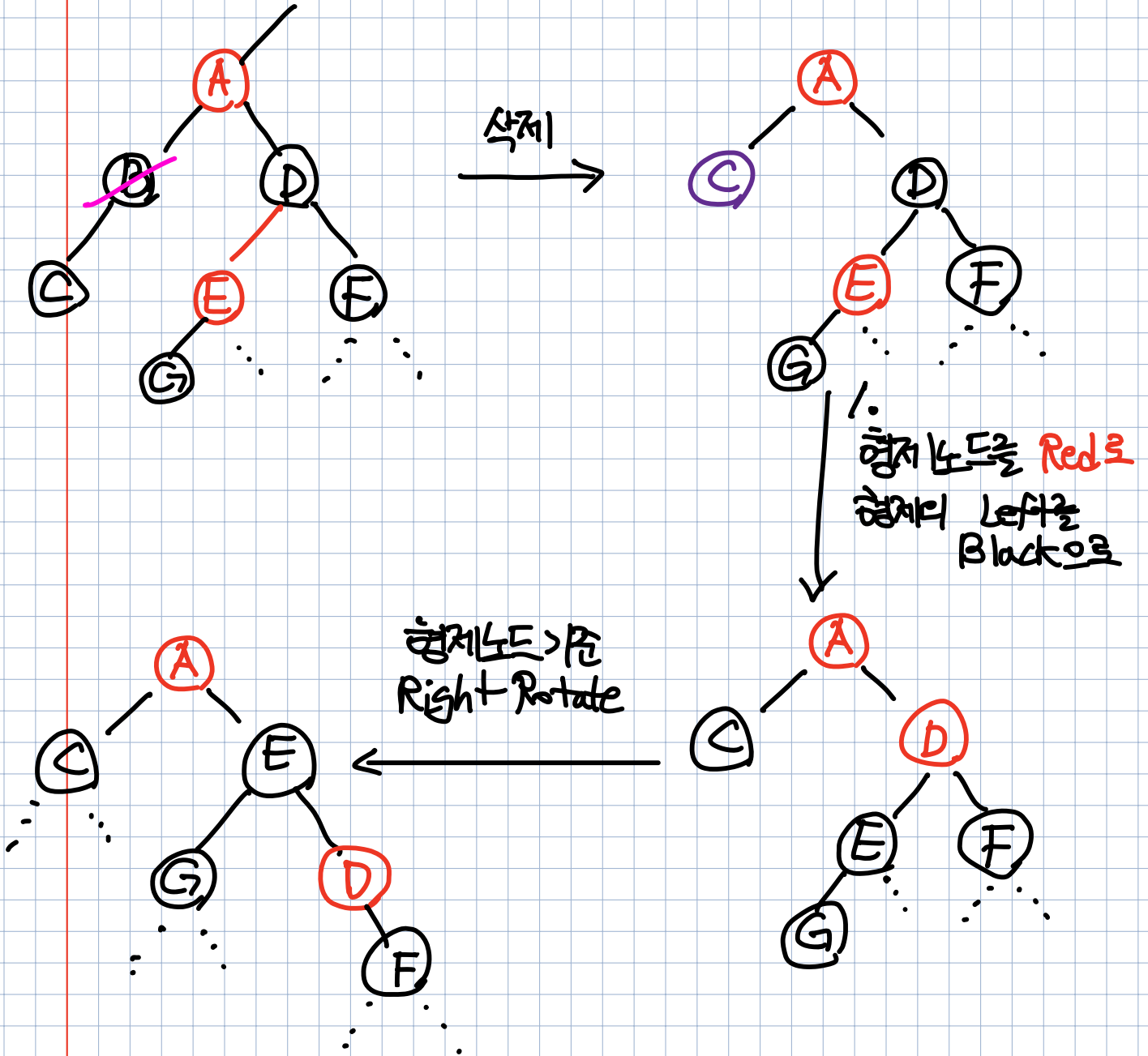


Title: 2. 형제가 검정색인 경우

Name: _____

B, 형제의 왼쪽 = Red, 오른쪽 = Black인 경우

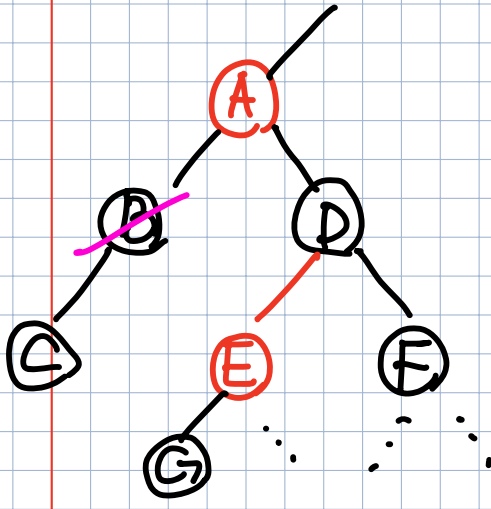
Date: _____



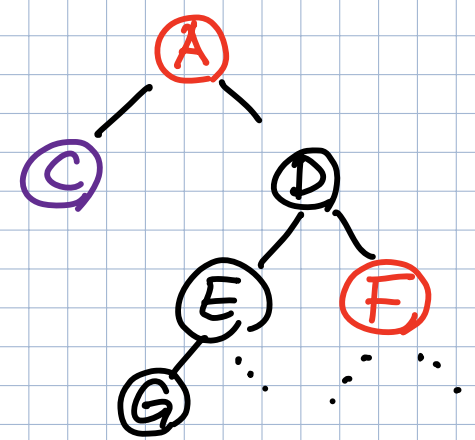
Title: 형제가 검정색인 경우
C. 형제의 오른쪽 = Red 인 경우

Name: _____

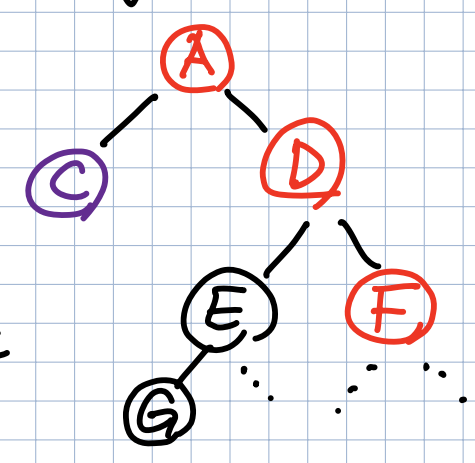
Date: _____



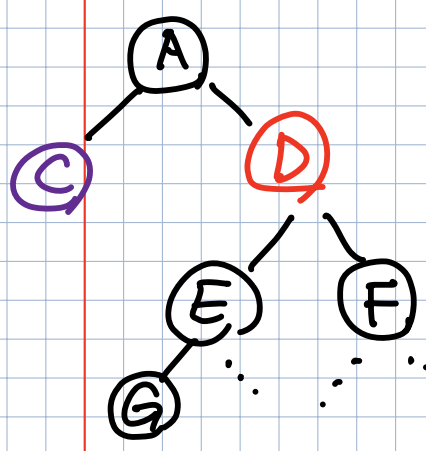
삭제 →



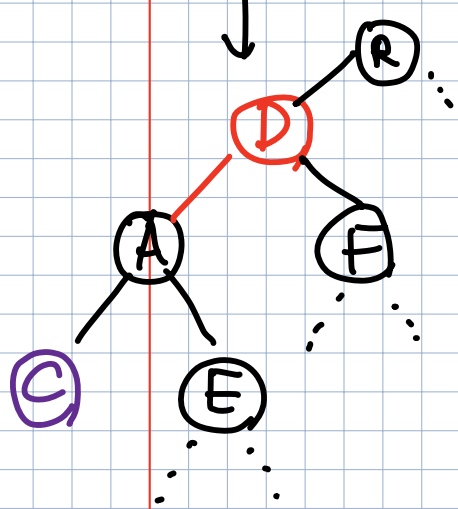
1. 부모노드의 색을 형제노드에 옮긴다.



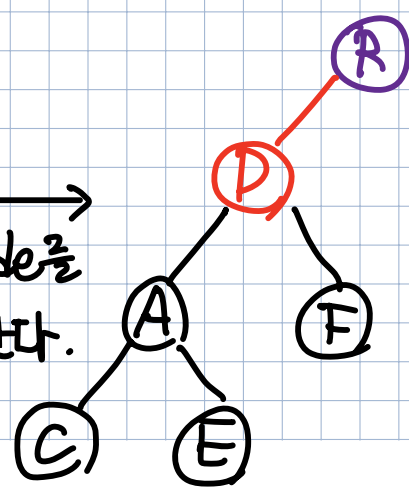
2. 부모노드와 형제의 오른쪽자식을 Black으로



3. 부모노드 > Root
Left Rotate



4. Root Node를 검정으로 칠한다.



Root는 이중흑색이므로
상관 X
~~~~~