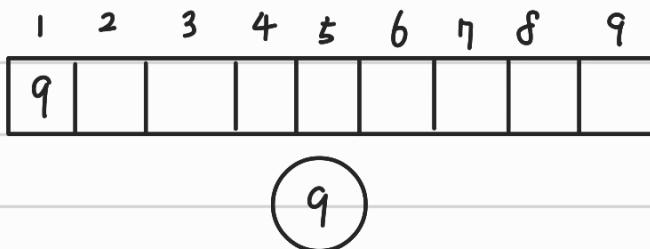


#1

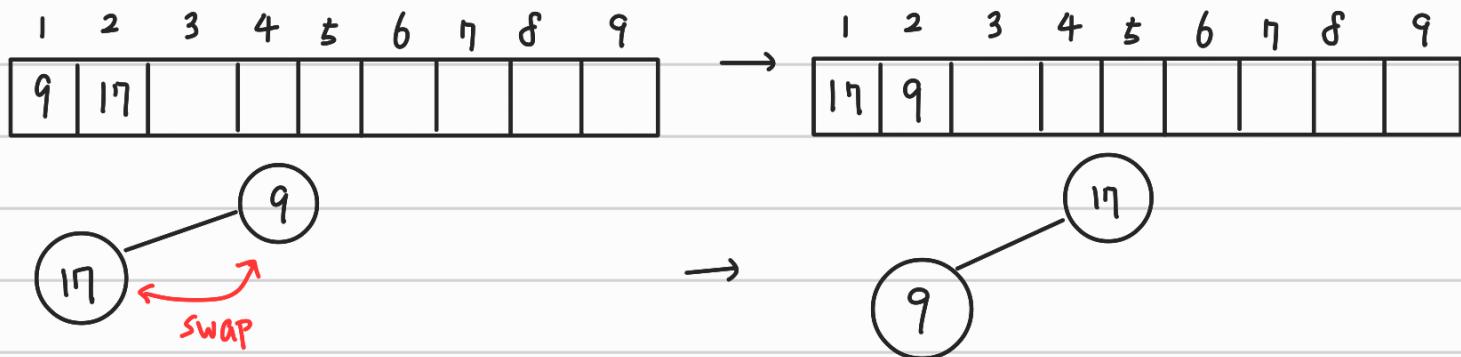
## 문제 1 (20점)

위에 주어진 순서대로 입력하여 Max-heap을 만들 경우 매 입력마다 Max-heap을 구성하는 array와 이 heap를 나타내는 개념적인 binary tree를 그리시오.

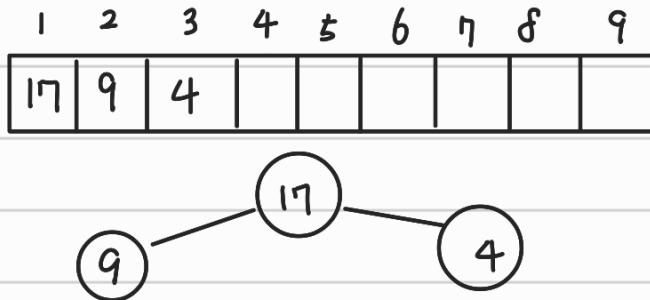
i) insert 9



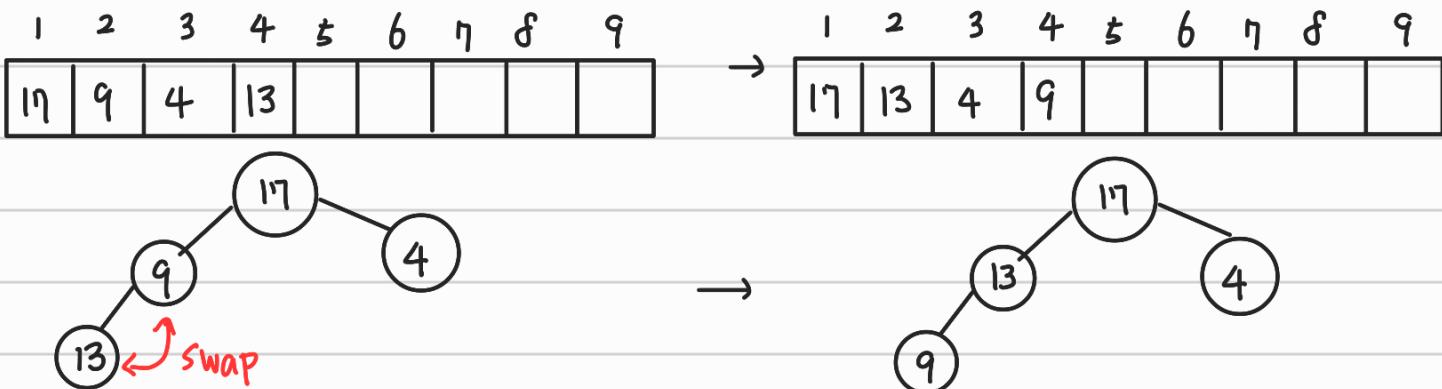
ii) insert 17



iii) insert 4

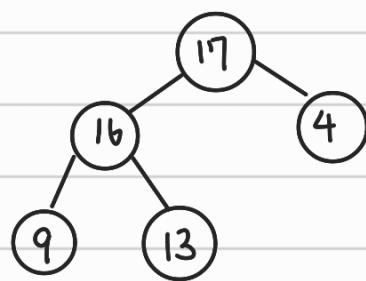
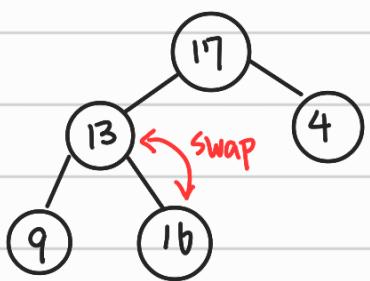


iv) insert 13



v) insert 16

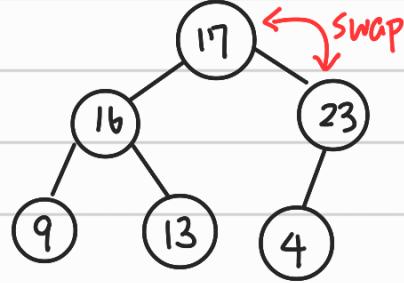
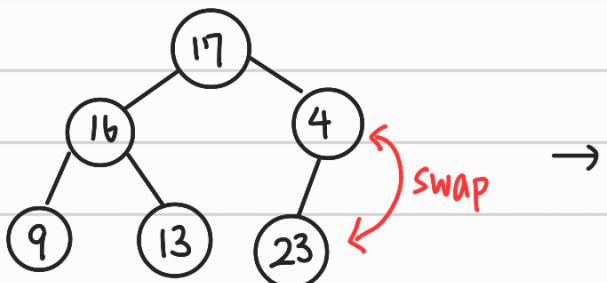




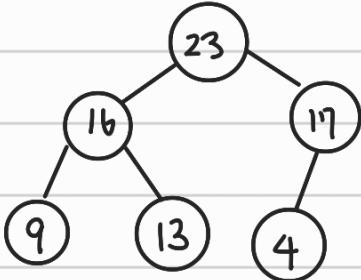
V.i) insert 23

1	2	3	4	5	6	7	8	9
17	16	4	9	13	23			

1	2	3	4	5	6	7	8	9
17	16	23	9	13	4			



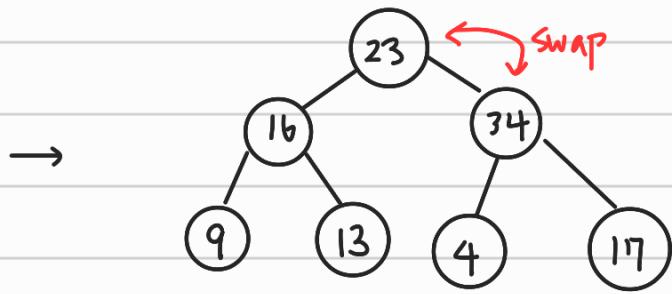
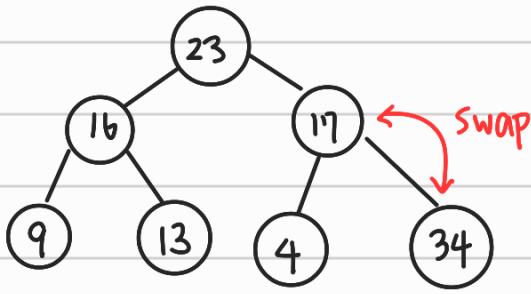
1	2	3	4	5	6	7	8	9
23	16	17	9	13	4			



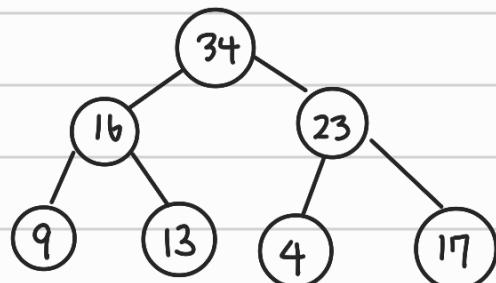
V.ii) insert 34

1	2	3	4	5	6	7	8	9
23	16	17	9	13	4	34		

1	2	3	4	5	6	7	8	9
23	16	34	9	13	4	17		



1	2	3	4	5	6	7	8	9
34	16	23	9	13	4	17		

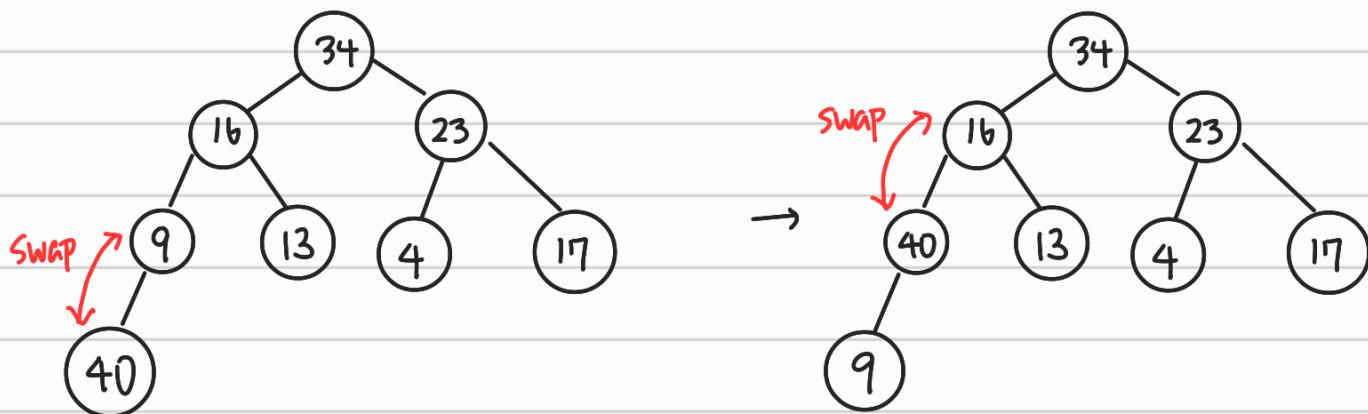


Viii) insert 40

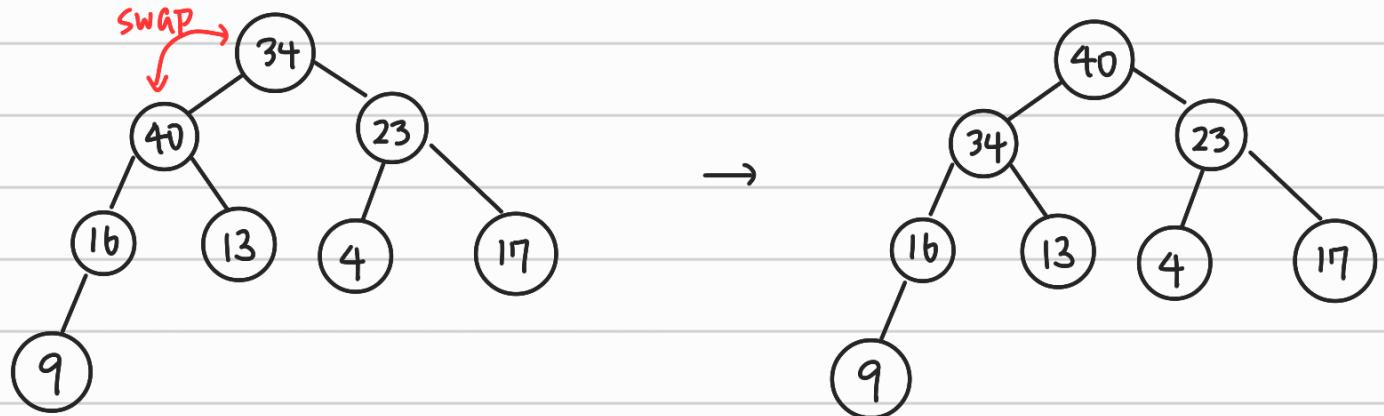
1	2	3	4	5	6	7	8	9
34	16	23	9	13	4	17	40	

→

1	2	3	4	5	6	7	8	9
34	16	23	40	13	4	17	9	

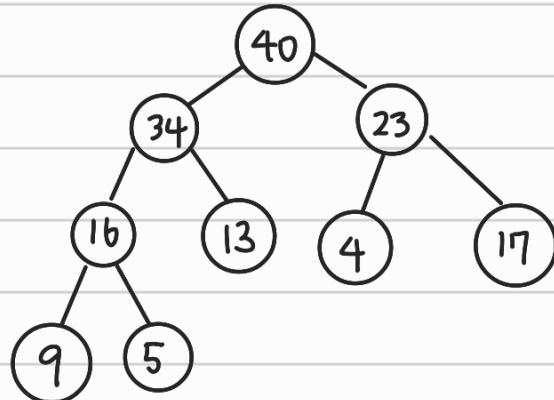


1	2	3	4	5	6	7	8	9
34	40	23	16	13	4	17	9	
→	40	34	23	16	13	4	17	9



ix) insert 5

1	2	3	4	5	6	7	8	9
40	34	23	16	13	4	17	9	5



#2

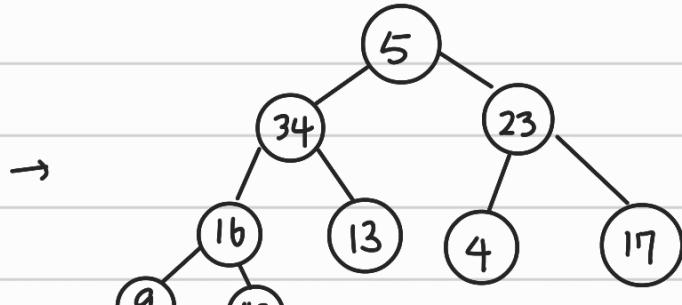
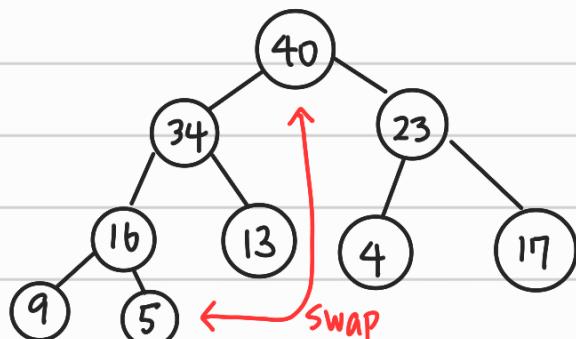
## 문제 2 (20점)

문제 1에서 만들어진 최종 Max-heap에 MAX-HEAP-EXTRACT-MAX 연산을 세번 연달아서 적용한다. 연산을 적용할 때마다 Max-heap을 구성하는 array와 이 heap를 나타내는 개념적인 binary tree를 그리시오.

① 1<sup>st</sup> trial

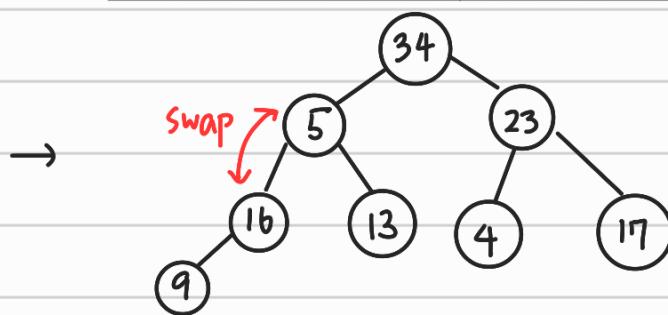
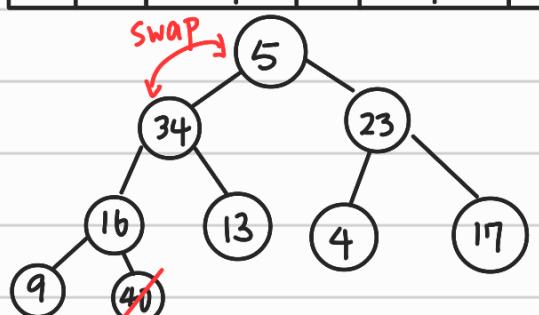
1	2	3	4	5	6	7	8	9
40	34	23	16	13	4	17	9	5

1	2	3	4	5	6	7	8	9
5	34	23	16	13	4	17	9	40



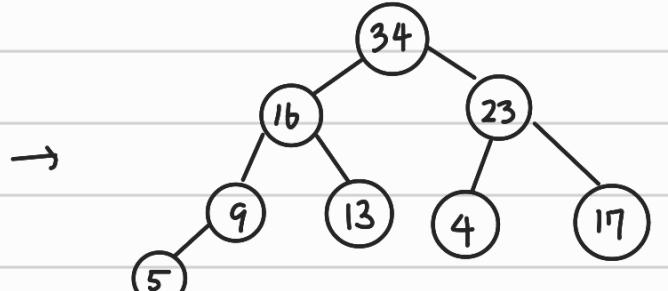
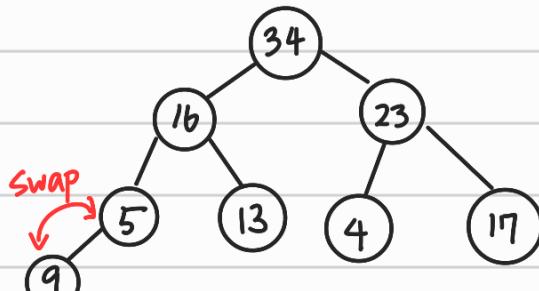
1	2	3	4	5	6	7	8	9
5	34	23	16	13	4	17	9	

1	2	3	4	5	6	7	8	9
34	5	23	16	13	4	17	9	



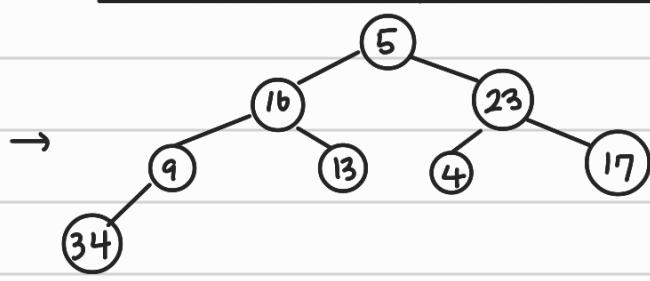
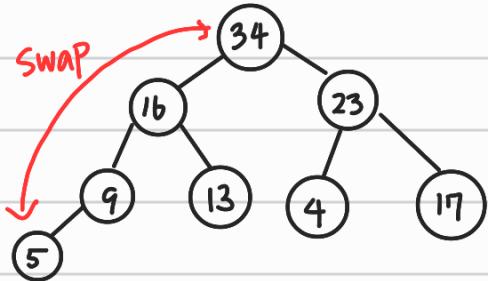
1	2	3	4	5	6	7	8	9
34	16	23	5	13	4	17	9	

1	2	3	4	5	6	7	8	9
34	16	23	9	13	4	17	5	

② 2<sup>nd</sup> trial

1	2	3	4	5	6	7	8	9
34	16	23	9	13	4	17	5	

1	2	3	4	5	6	7	8	9
5	16	23	9	13	4	17	34	

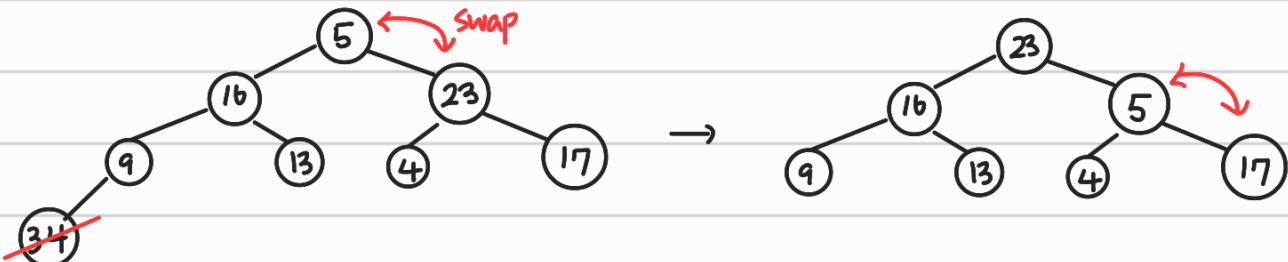


→

1	2	3	4	5	6	7	8	9
5	16	23	9	13	4	17		

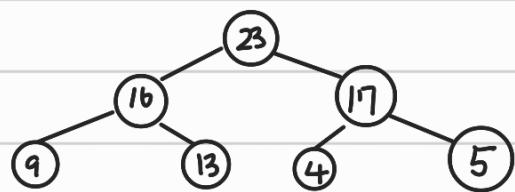
→

1	2	3	4	5	6	7	8	9
23	16	5	9	13	4	17		



→

1	2	3	4	5	6	7	8	9
23	16	17	9	13	4	5		

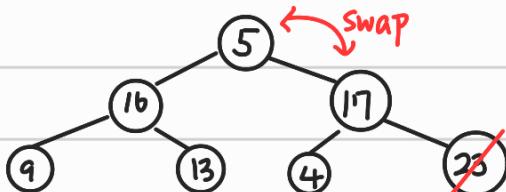
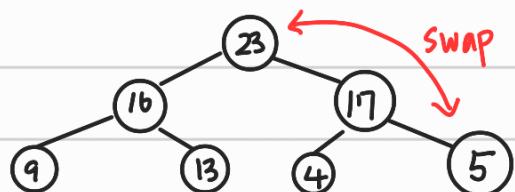


③ 3<sup>rd</sup> trial

1	2	3	4	5	6	7	8	9
23	16	17	9	13	4	5		

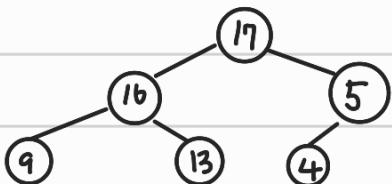
→

1	2	3	4	5	6	7	8	9
5	16	17	9	13	4	23		



→

1	2	3	4	5	6	7	8	9
17	16	5	9	13	4			



# 3

### 문제 3 (20점)

위에 주어진 key 순서대로 입력하여 binary search tree를 만들 경우 매 입력마다 만들어진 binary search tree를 그리시오.

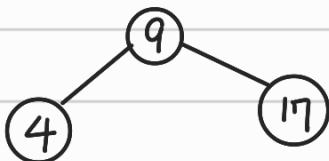
i) insert 9



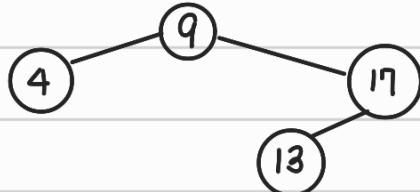
ii) insert 17



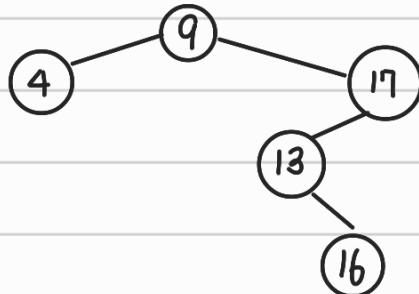
iii) insert 4



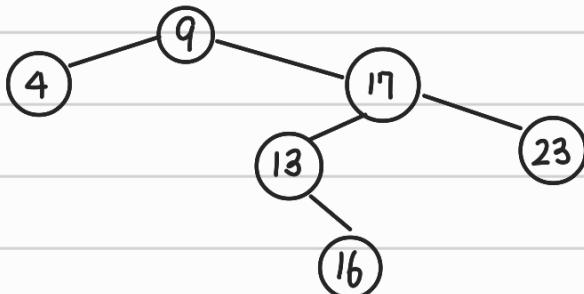
iv) insert 13



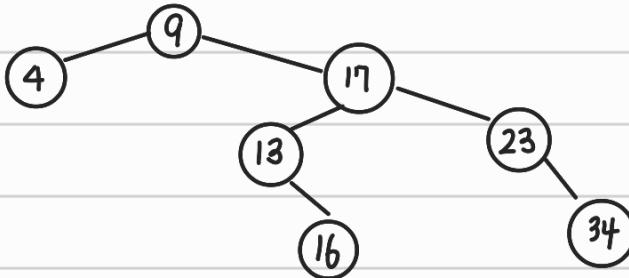
v) insert 16



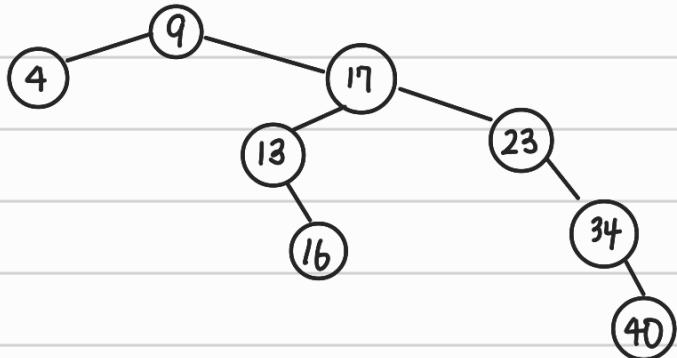
vi) insert 23



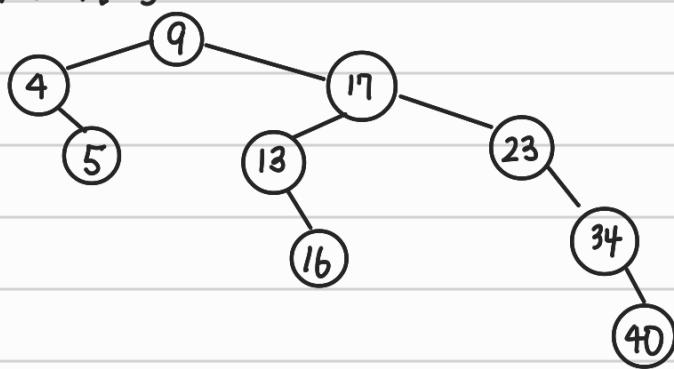
vii) insert 34



viii) insert 40



ix) insert 5

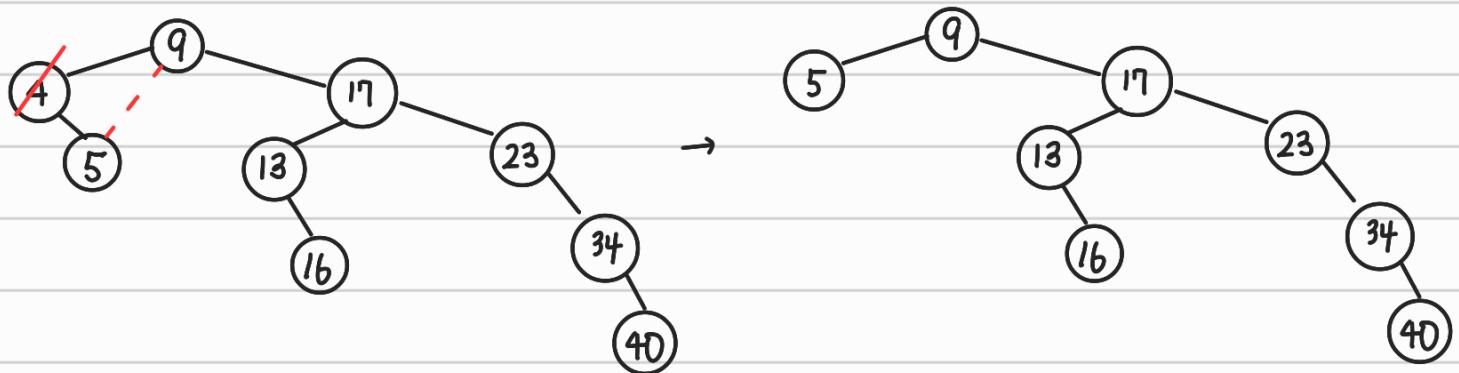


## #4 문제 4 (20점)

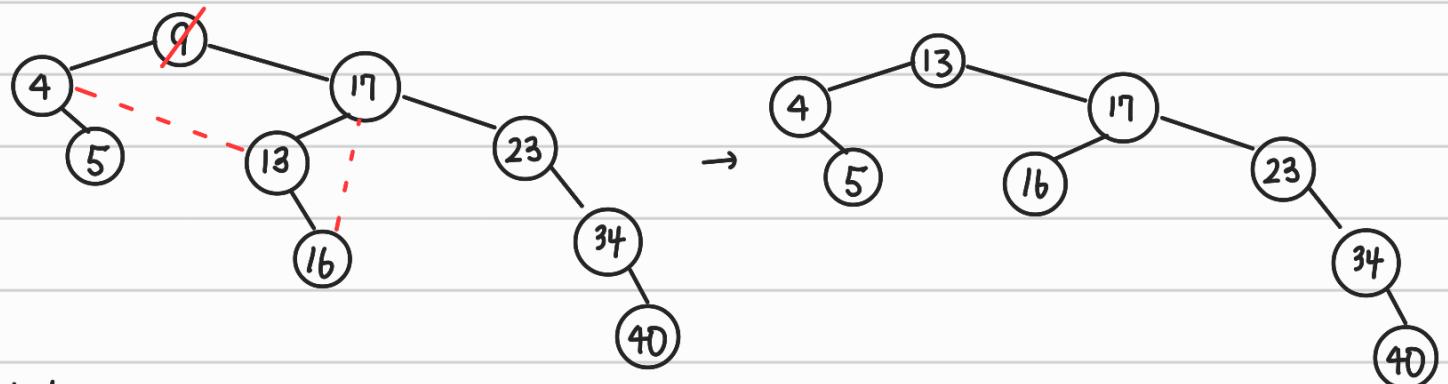
문제 3에서 만들어진 최종 BST에 아래와 같은 key를 가진 노드를 제거하는 TREE-DELETE 연산을 적용할 때 만들어지는 BST를 그리시오 (연달아서 차례대로 적용하는 것이 아니다).

- (a) 4
- (b) 9
- (c) 34
- (d) 16
- (e) 17

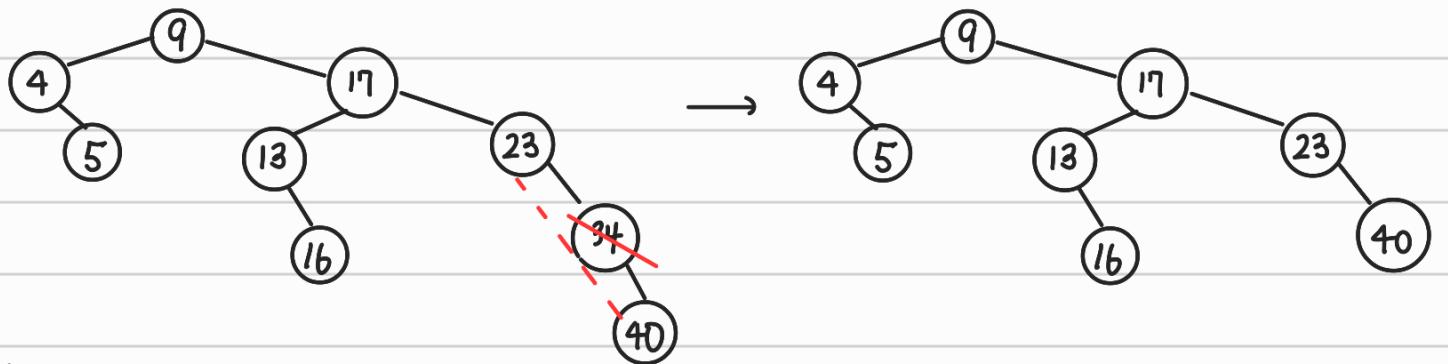
(a) delete 4



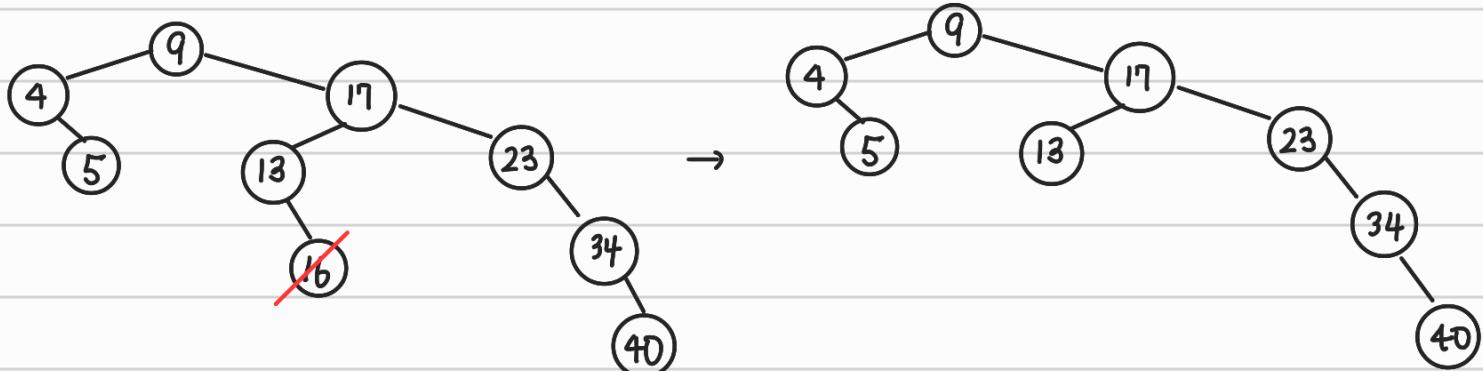
(b) delete 9



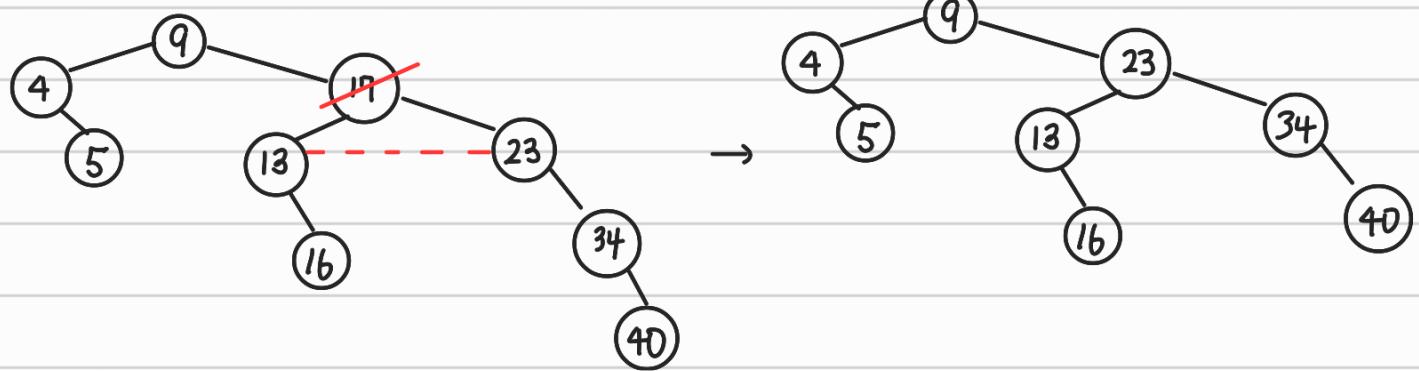
(c) delete 34



(d) delete 16



(e) delete 17



## #5. 문제 5 (20점)

위에 주어진 key 순서대로 입력하여 red-black tree를 만들 경우 매 입력마다 만들어진 red-black tree를 그리시오. 또, 각 입력 과정마다 강의에서 설명한 case 세 개 중 해당되는 case를 모두 적으시오.

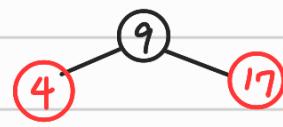
i) insert 9



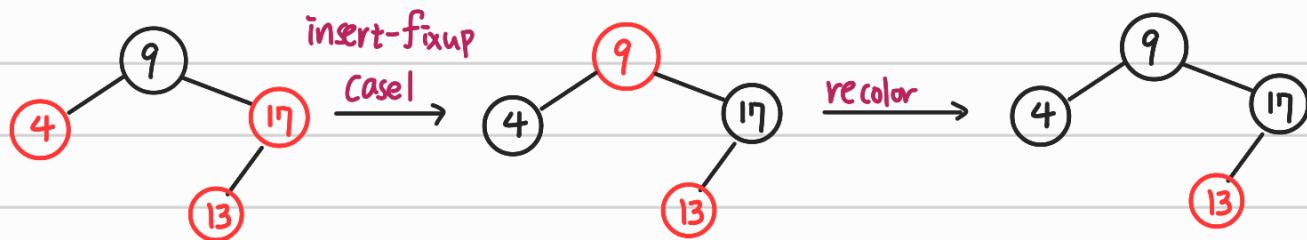
ii) insert 17



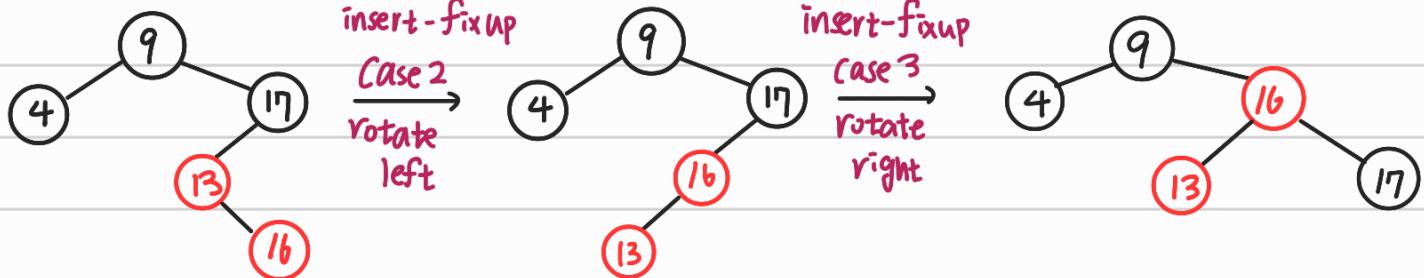
iii) insert 4



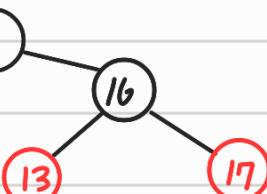
iv) insert 13



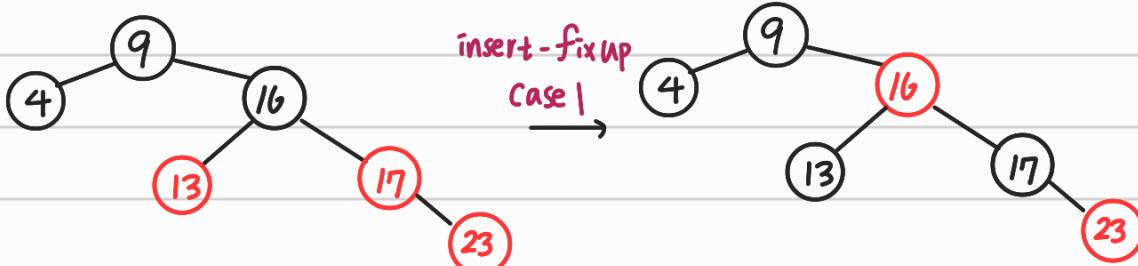
v) insert 16



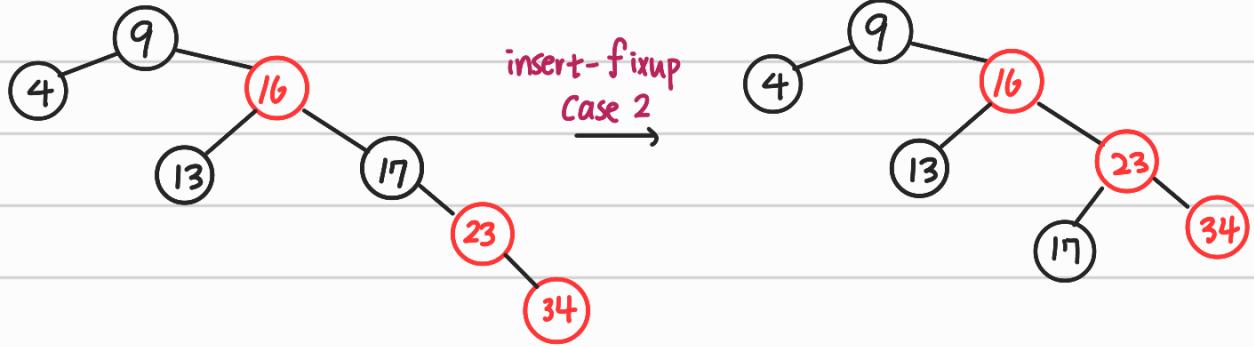
recolor →



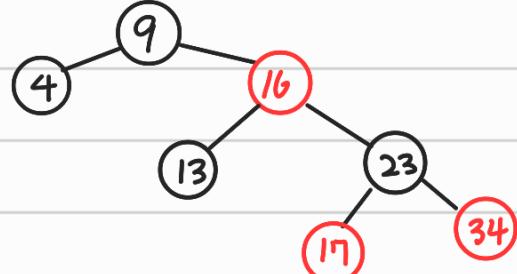
vi) insert 23



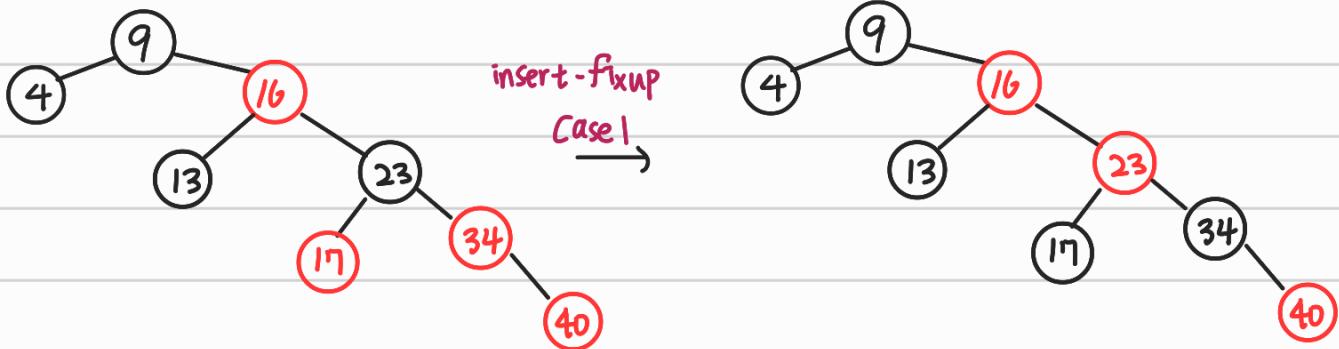
Vii) insert 34



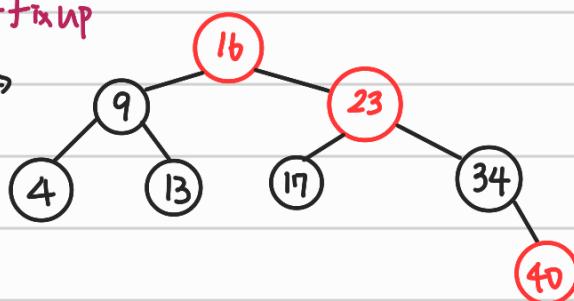
recolor →



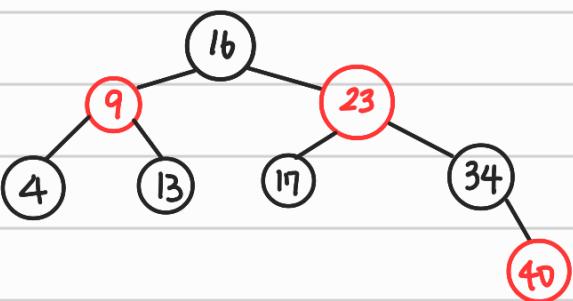
Viii) insert 40



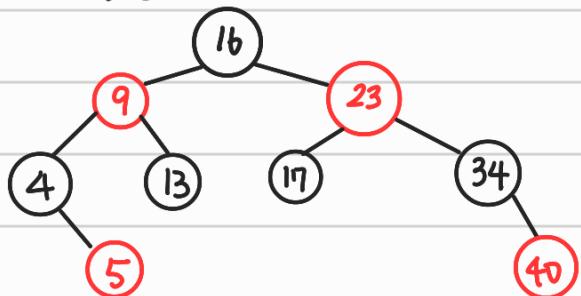
insert-fixup  
Case 2 →



recolor →



ix) insert 5



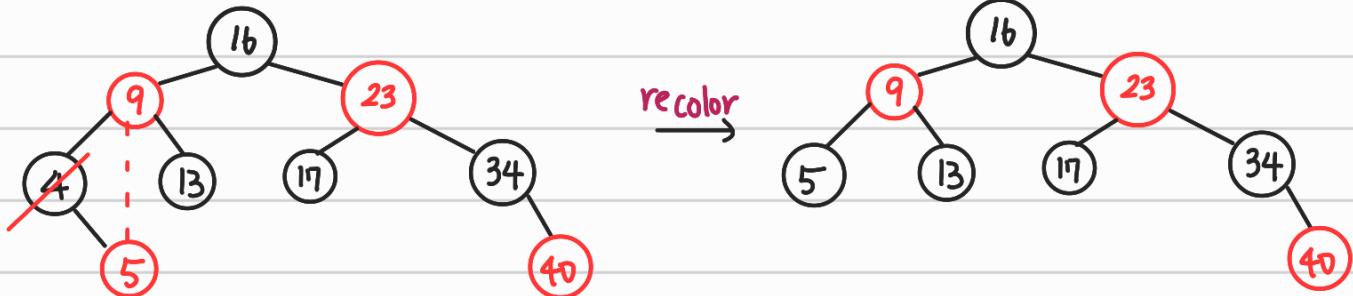
#6

## 문제 6 (20점)

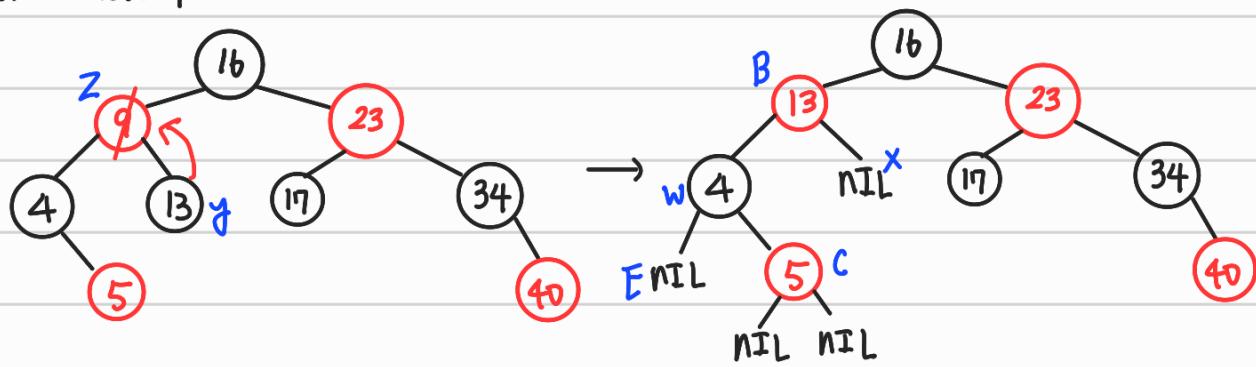
문제 5에서 만들어진 최종 RBT에 아래와 같은 key를 가진 노드를 제거하는 RB-DELETE 연산을 적용할 때 만들어지는 BST를 그리시오 (연달아서 차례대로 적용하는 것이 아니다). 또, 각 제거 과정마다 강의에서 설명한 case 네 개 중 해당되는 case를 모두 적으시오.

- (a) 4
- (b) 9
- (c) 34
- (d) 16
- (e) 17

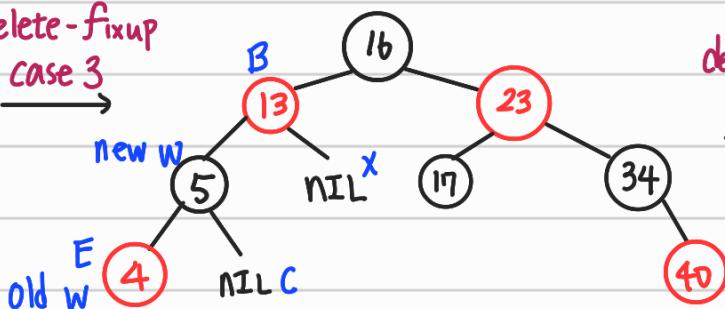
(a) delete 4



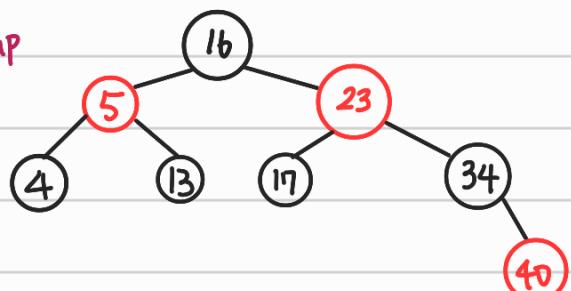
(b) delete 9



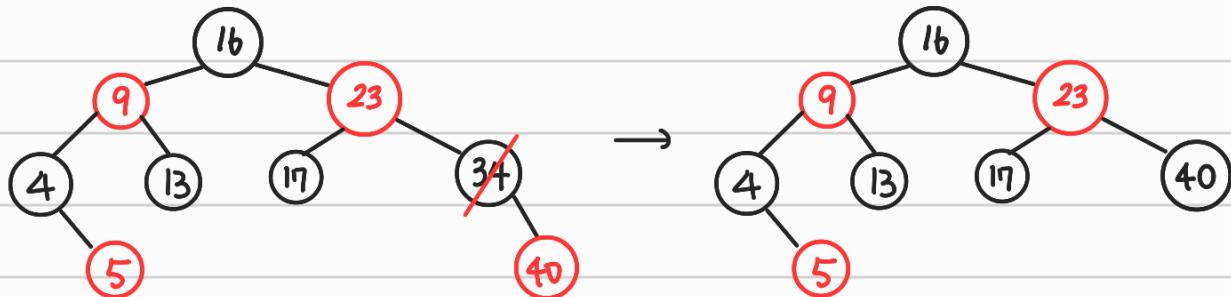
delete-fixup  
case 3



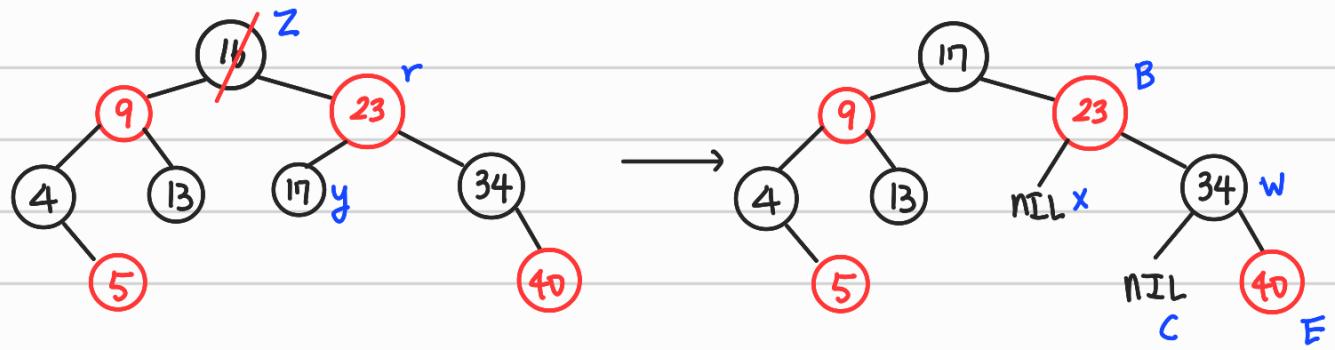
delete-fixup  
case 4



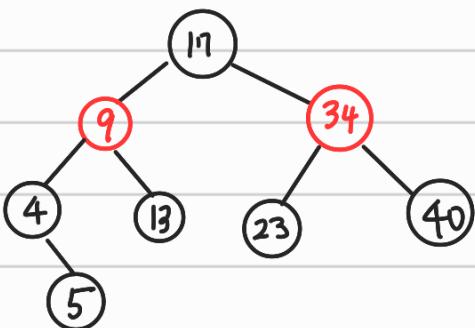
(c) delete 34



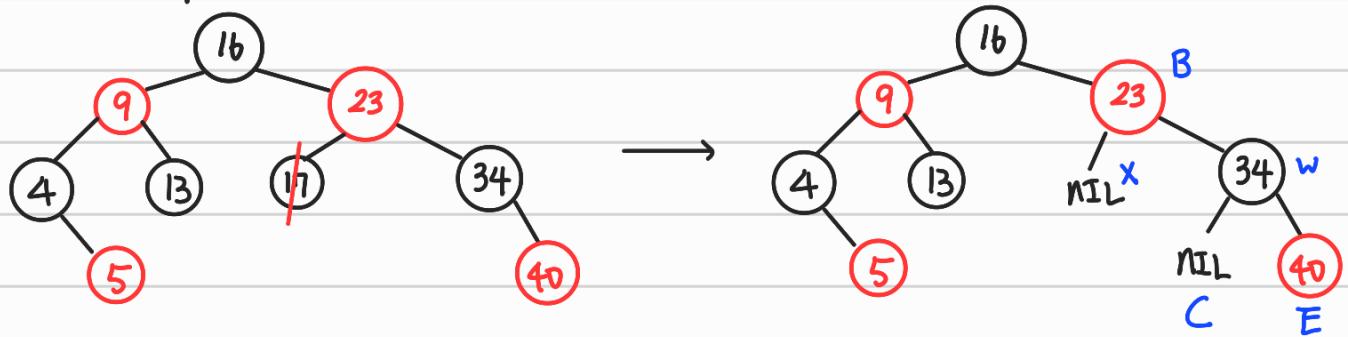
(d) delete 16



delete-fixup  
case 4



(e) delete 17



delete-fixup  
case 4

