

컴퓨터공학 All in One

C/C++ 문법, 자료구조 및 심화 프로젝트 (나동빈)
제 83강 - B 트리의 개요와 알고리즘 원리

B 트리의 개요와 알고리즘 원리

B 트리의 개요

- B Tree는 AVL 트리와 같은 균형 트리(Balanced Tree)입니다.
- B Tree는 자식을 두 개만 가지는 이진 트리의 단점을 보완할 수 있습니다.

B 트리의 개요와 알고리즘 원리

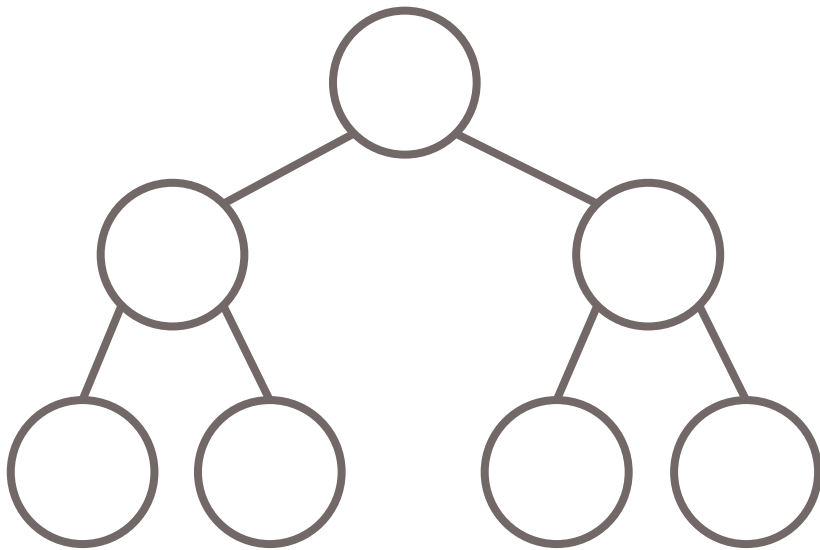
균형 트리(Balanced Tree)

- 균형 트리의 종류로는 AVL 트리, 2-3 트리, 2-3-4 트리 등이 있습니다.
- B Tree를 이해하면 2-3 트리, 2-3-4 트리의 원리를 쉽게 이해할 수 있습니다.
- 데이터베이스들은 매우 많은 데이터를 포함한다는 점에서 B Tree 구조를 채택하고 있습니다.

B 트리의 개요와 알고리즘 원리

자식의 개수에 따른 메모리 소모

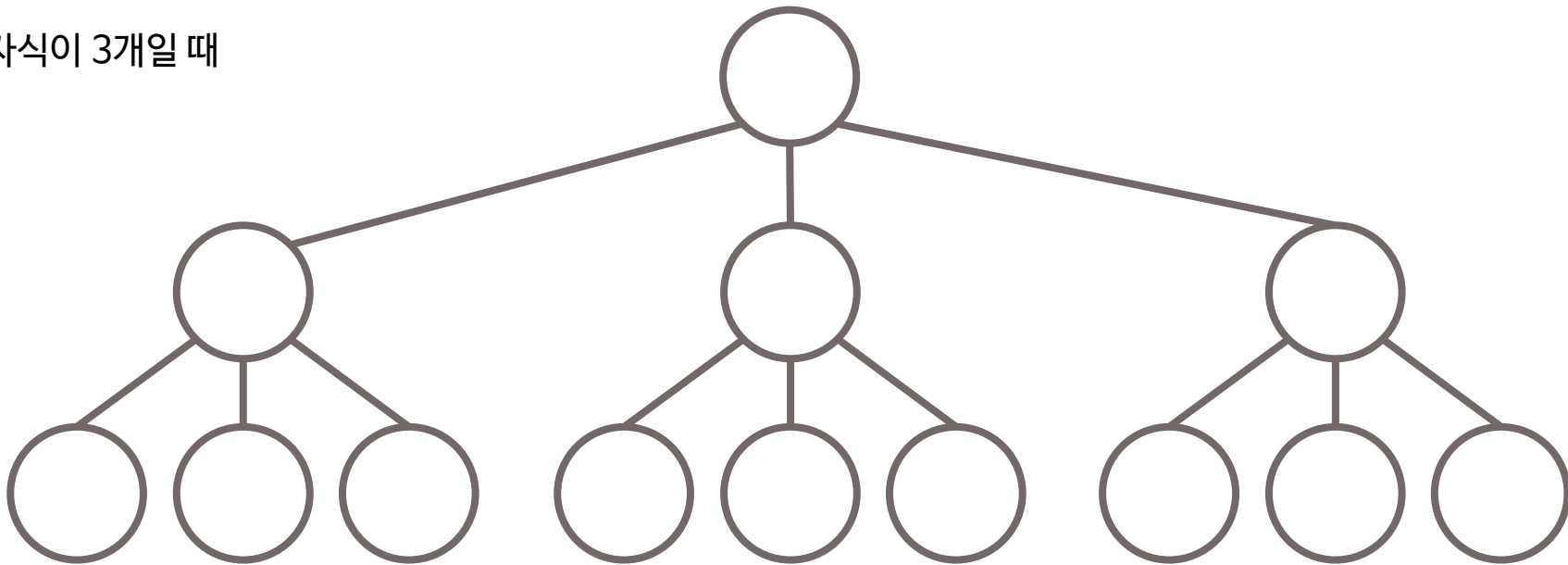
- 자식이 2개일 때



B 트리의 개요와 알고리즘 원리

자식의 개수에 따른 메모리 소모

- 자식이 3개일 때



B 트리의 개요와 알고리즘 원리

자식의 개수에 따른 메모리 소모

- 자식의 개수가 많아질수록 추가적으로 소모되는 메모리 용량이 많습니다.
- 자식의 개수가 많아질수록 빠르게 데이터에 접근할 가능성이 높습니다.

B 트리의 개요와 알고리즘 원리

B 트리의 특징

- B Tree는 AVL 트리와 같은 균형 트리(Balanced Tree)입니다.
- B Tree는 삽입 및 삭제 이후에도 균형 트리를 유지할 수 있습니다.
- B Tree는 자식을 두 개만 가지는 이진 트리의 단점을 보완할 수 있습니다.
- 매우 많은 데이터를 포함하는 디스크 기반 솔루션으로 설계된 자료구조입니다.

B 트리의 개요와 알고리즘 원리

B 트리의 규칙

- 노드의 데이터 수가 최대 N 이라면, 자식의 수는 최대 $N + 1$ 입니다.
- 각 노드의 데이터는 정렬된 상태입니다.
- 리프 노드(Leaf Node)는 모두 같은 레벨에 있으므로 완전한 균형 트리 형태를 유지합니다.
- 특정한 데이터의 왼쪽 서브 트리는 해당 데이터보다 작으며, 오른쪽 서브 트리는 해당 데이터보다 큼니다.
- 루트(Root) 노드가 자식을 가질 때는 최소한 2개 이상의 자식을 가집니다.
- 루트(Root)와 리프(Leaf) 노드를 제외한 노드는 최소한 $\{(\text{차수}) / 2\}$ 의 올림 만큼의 자식을 가지고 있습니다.
- 데이터에 중복이 없다고 가정합니다.

B 트리의 개요와 알고리즘 원리

B Tree 삽입 예제 (Degree = 3)



※ 7 삽입 ※

B 트리의 개요와 알고리즘 원리

B Tree 삽입 예제 (Degree = 3)



※ 5 삽입 ※

B 트리의 개요와 알고리즘 원리

B Tree 삽입 예제 (Degree = 3)

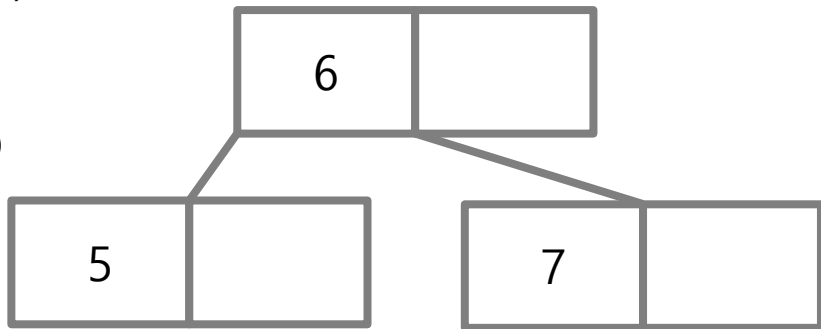


※ 6 삽입 ※ (데이터의 개수 3개 도달)

B 트리의 개요와 알고리즘 원리

B Tree 삽입 예제 (Degree = 3)

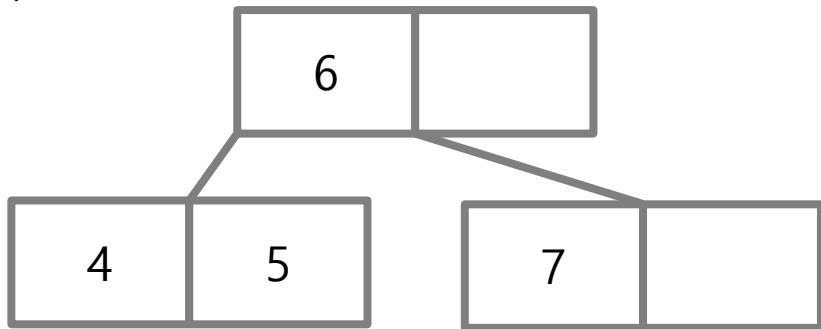
※ 6 삽입 ※ (6을 위로 올리기)



B 트리의 개요와 알고리즘 원리

B Tree 삽입 예제 (Degree = 3)

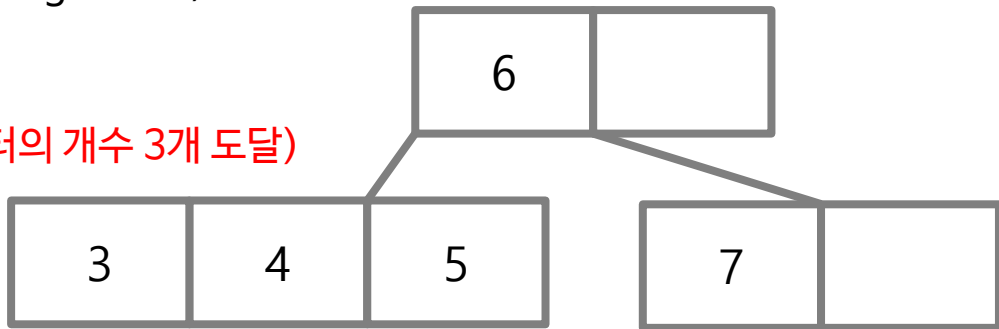
※ 4 삽입 ※



B 트리의 개요와 알고리즘 원리

B Tree 삽입 예제 (Degree = 3)

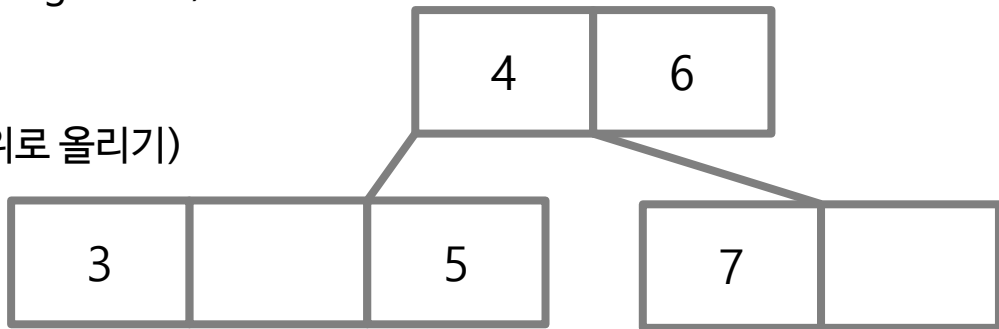
※ 3 삽입 ※ (데이터의 개수 3개 도달)



B 트리의 개요와 알고리즘 원리

B Tree 삽입 예제 (Degree = 3)

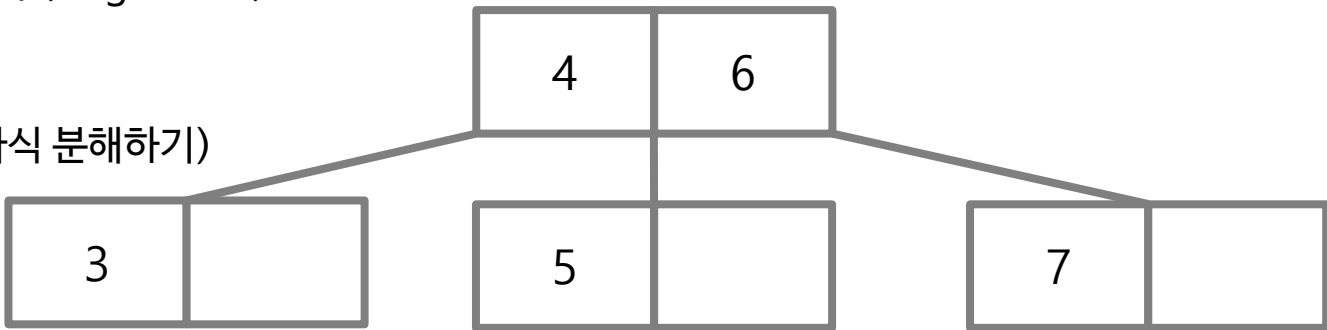
※ 3 삽입 ※ (4을 위로 올리기)



B 트리의 개요와 알고리즘 원리

B Tree 삽입 예제 (Degree = 3)

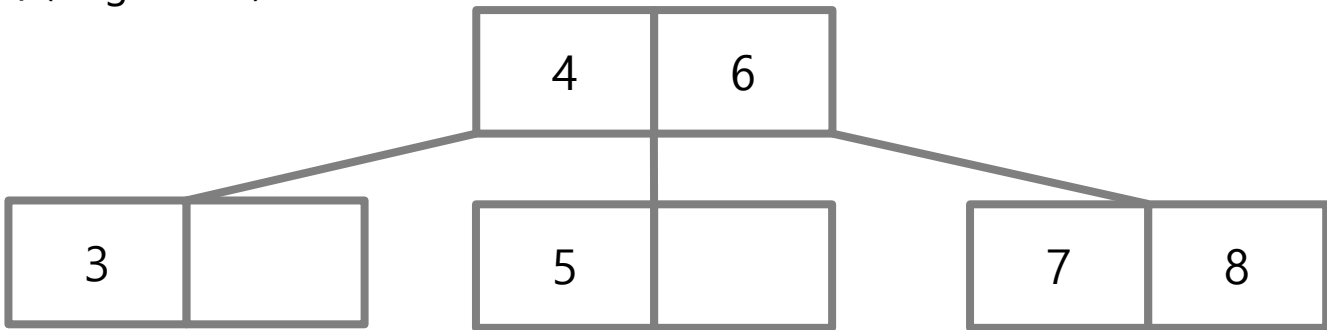
※ 3 삽입 ※ (자식 분해하기)



B 트리의 개요와 알고리즘 원리

B Tree 삽입 예제 (Degree = 3)

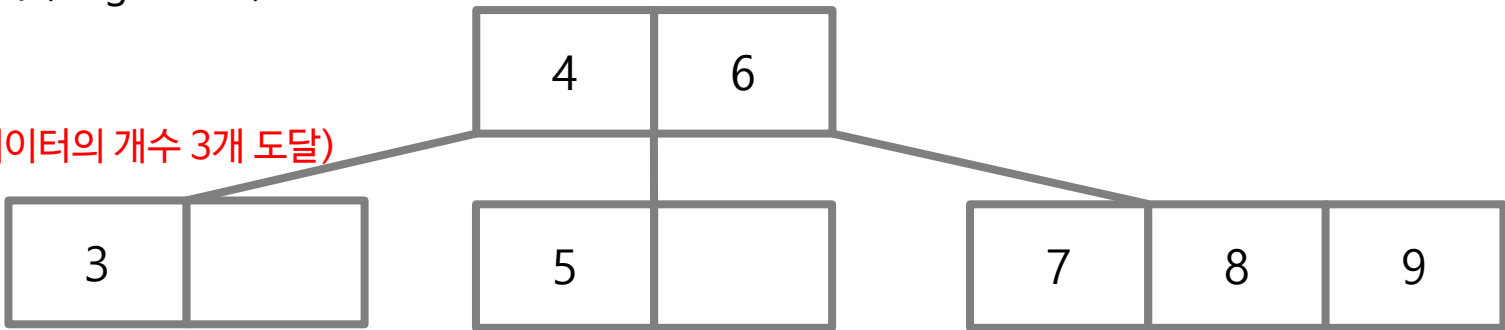
※ 8 삽입 ※



B 트리의 개요와 알고리즘 원리

B Tree 삽입 예제 (Degree = 3)

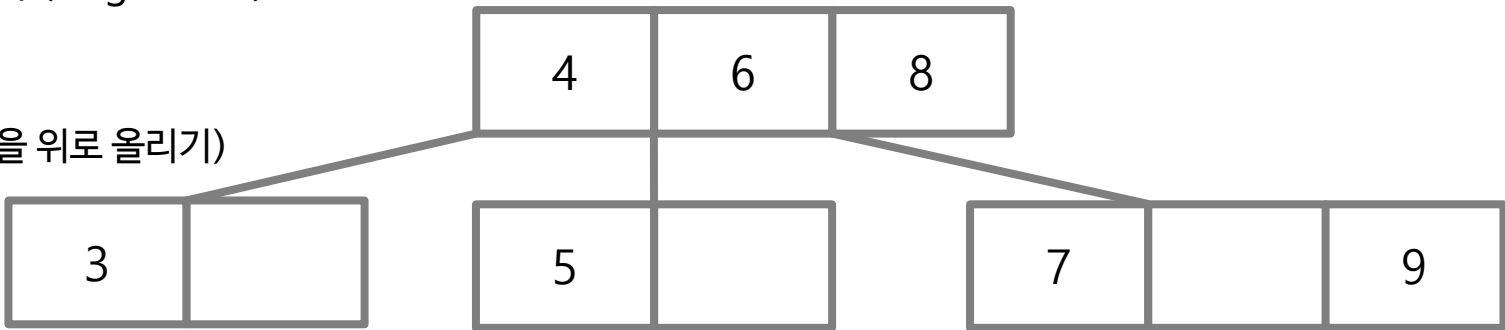
※ 9 삽입 ※ (데이터의 개수 3개 도달)



B 트리의 개요와 알고리즘 원리

B Tree 삽입 예제 (Degree = 3)

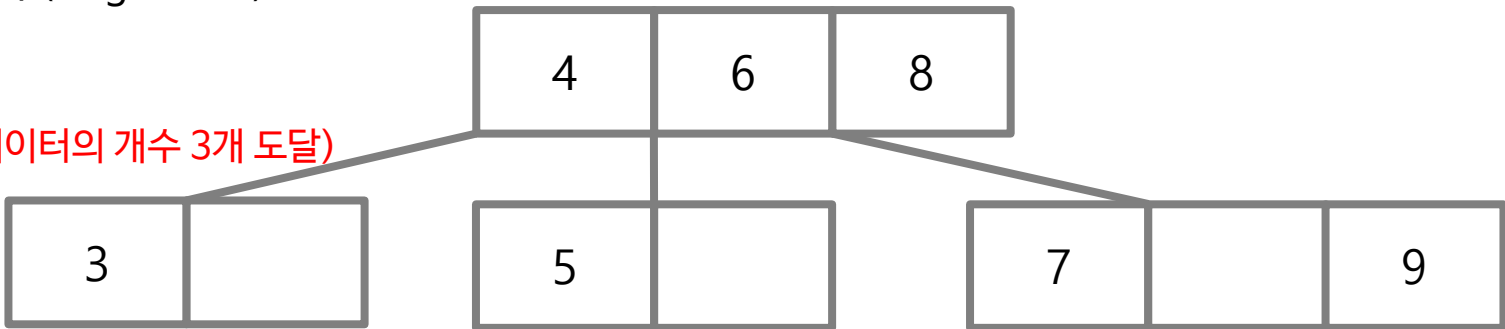
※ 9 삽입 ※ (8을 위로 올리기)



B 트리의 개요와 알고리즘 원리

B Tree 삽입 예제 (Degree = 3)

※ 9 삽입 ※ (데이터의 개수 3개 도달)

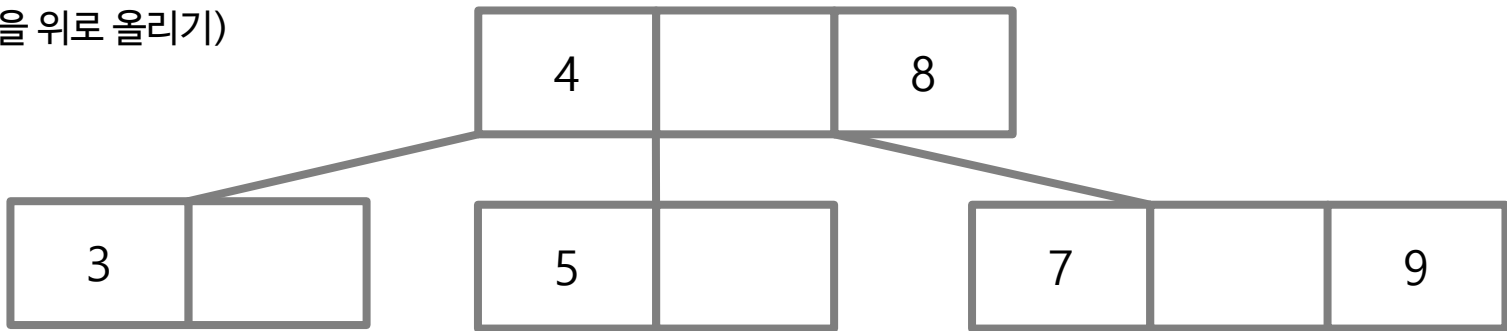


B 트리의 개요와 알고리즘 원리

B Tree 삽입 예제 (Degree = 3)



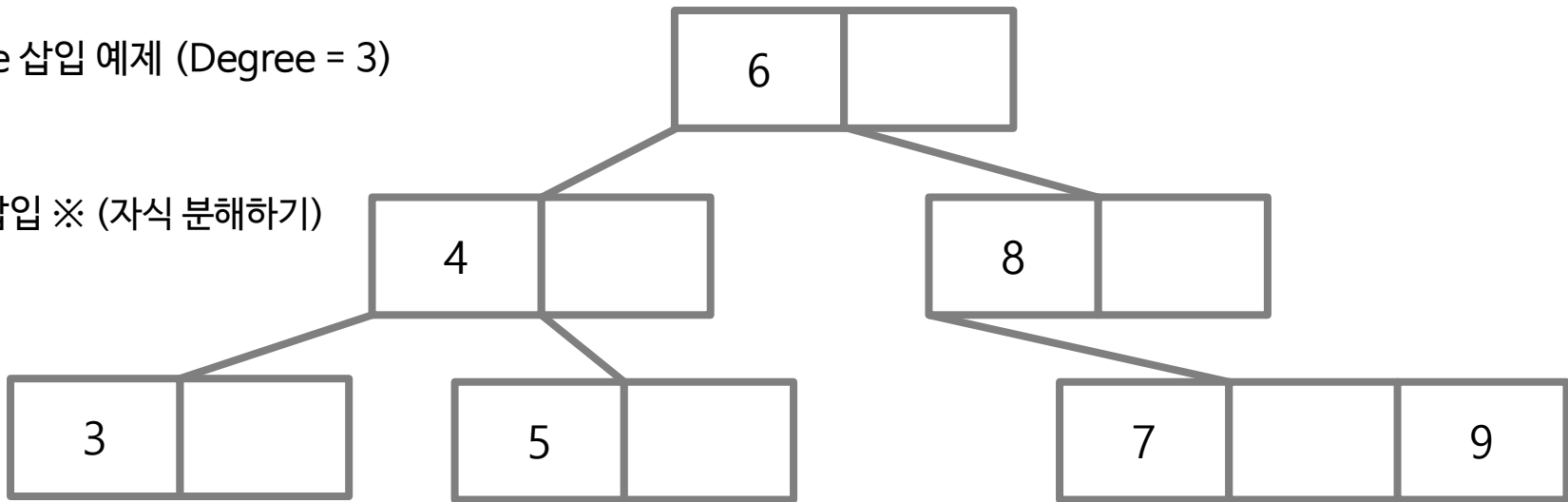
※ 9 삽입 ※ (6을 위로 올리기)



B 트리의 개요와 알고리즘 원리

B Tree 삽입 예제 (Degree = 3)

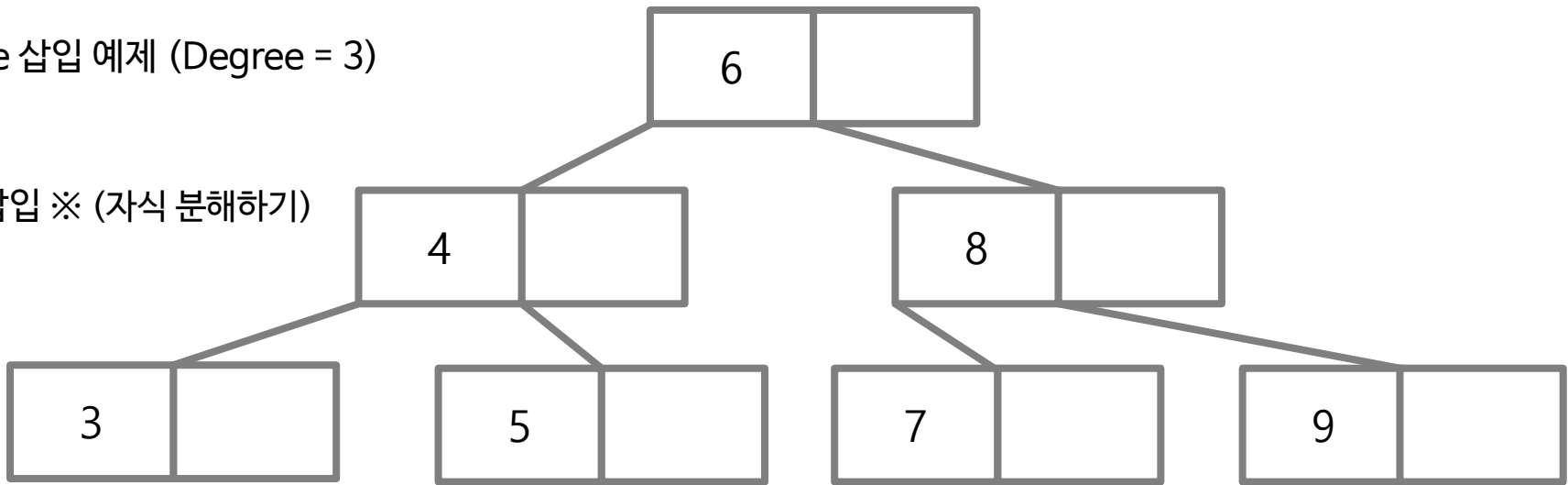
※ 9 삽입 ※ (자식 분해하기)



B 트리의 개요와 알고리즘 원리

B Tree 삽입 예제 (Degree = 3)

※ 9 삽입 ※ (자식 분해하기)



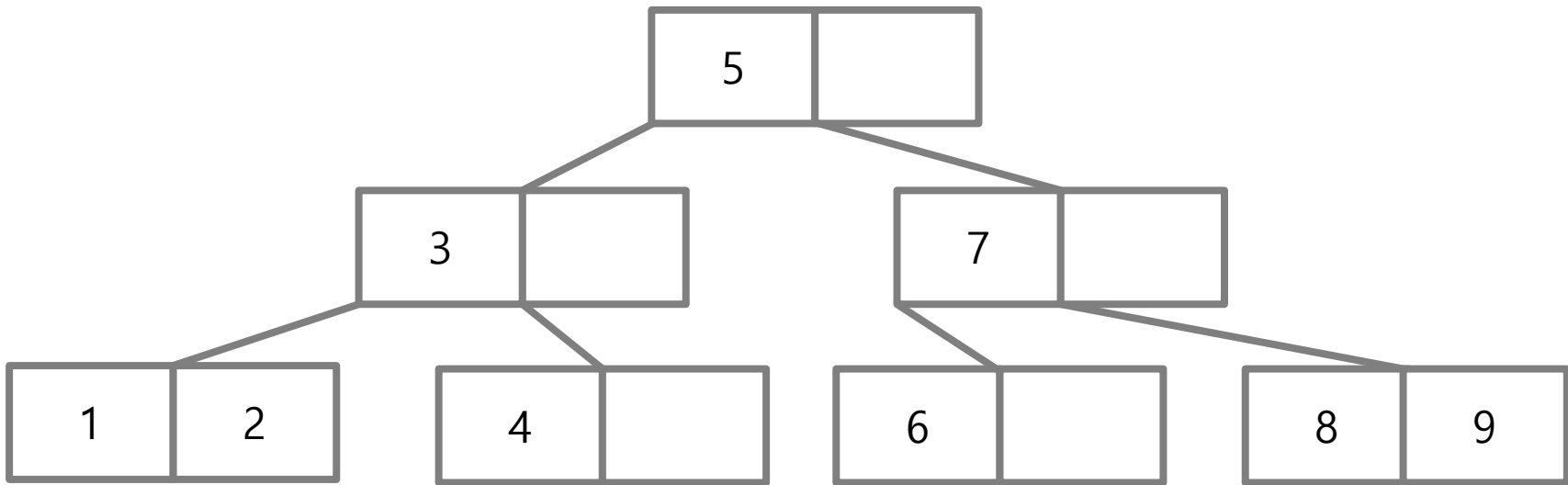
B 트리의 개요와 알고리즘 원리

B Tree 삭제 예제 (Degree = 3)

- B Tree의 삭제는 회전과 병합을 통해 이루어집니다.

B 트리의 개요와 알고리즘 원리

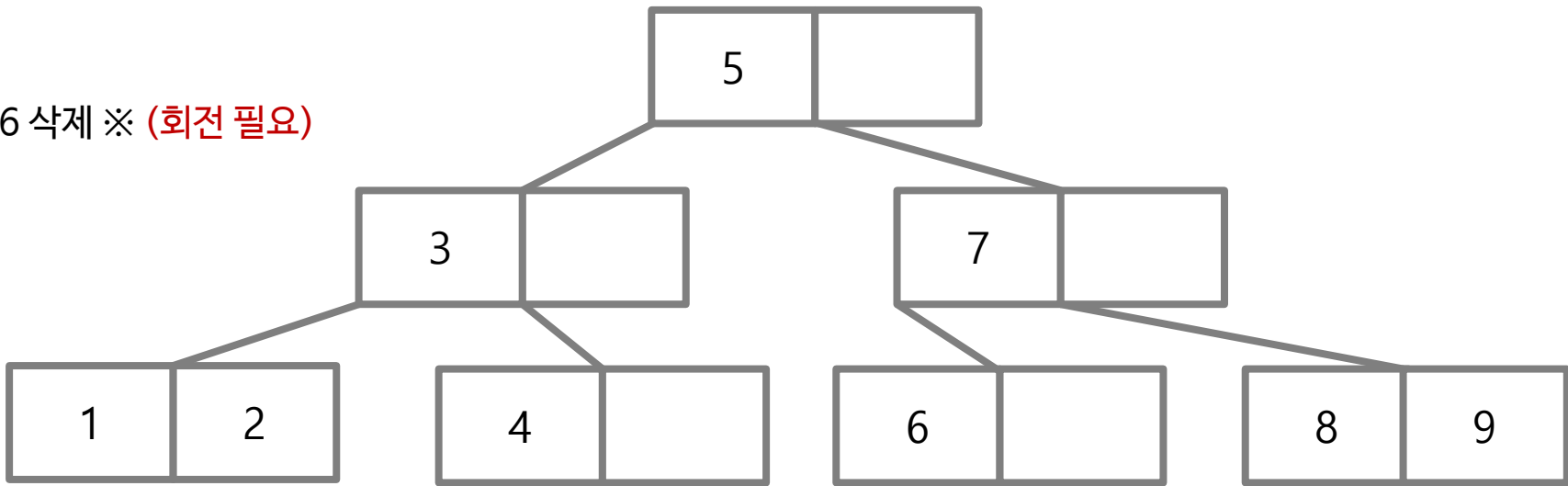
B Tree 삭제 예제 (Degree = 3)



B 트리의 개요와 알고리즘 원리

B Tree 삭제 예제 (Degree = 3)

※ 6 삭제 ※ (회전 필요)

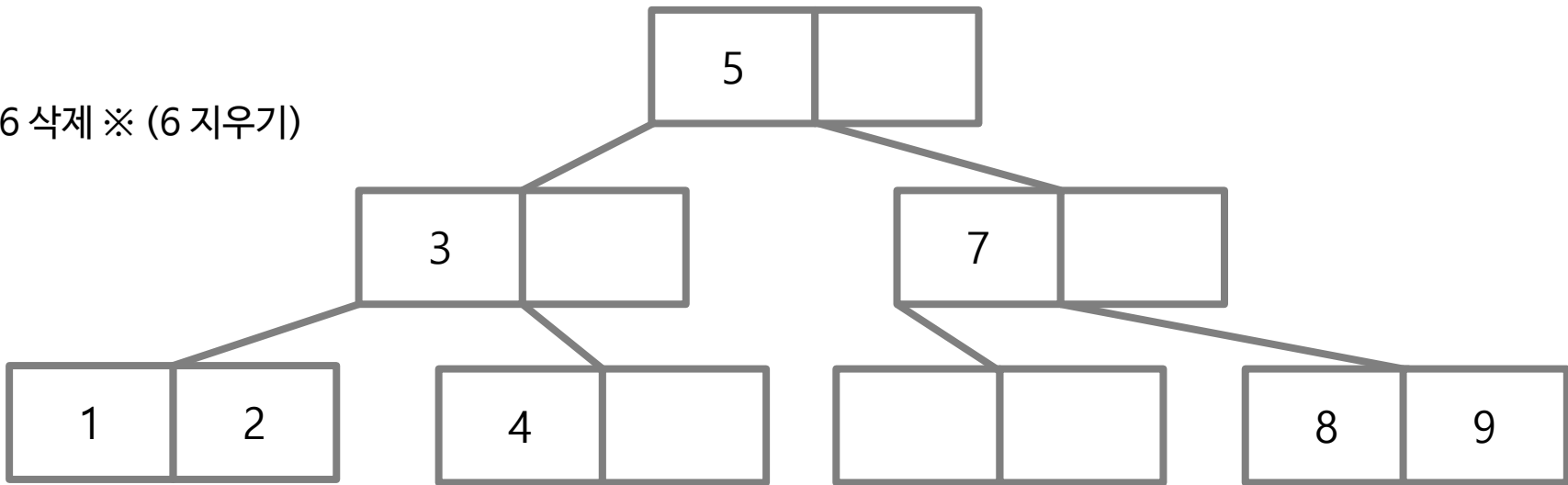


가져올 수 있음

B 트리의 개요와 알고리즘 원리

B Tree 삭제 예제 (Degree = 3)

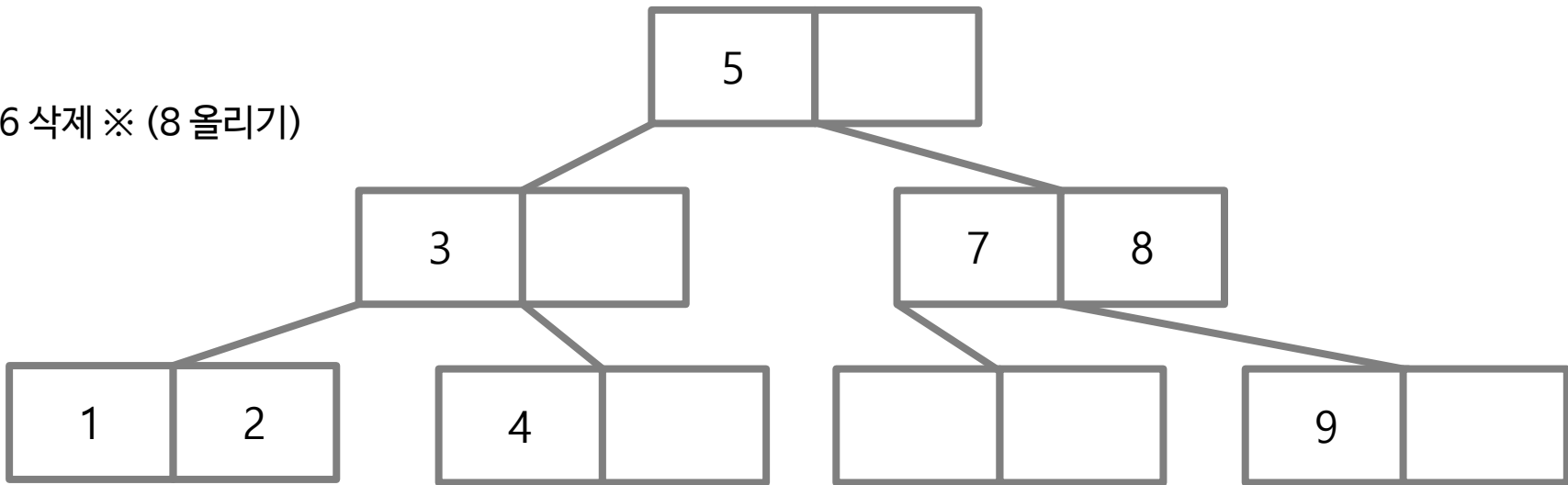
※ 6 삭제 ※ (6 지우기)



B 트리의 개요와 알고리즘 원리

B Tree 삭제 예제 (Degree = 3)

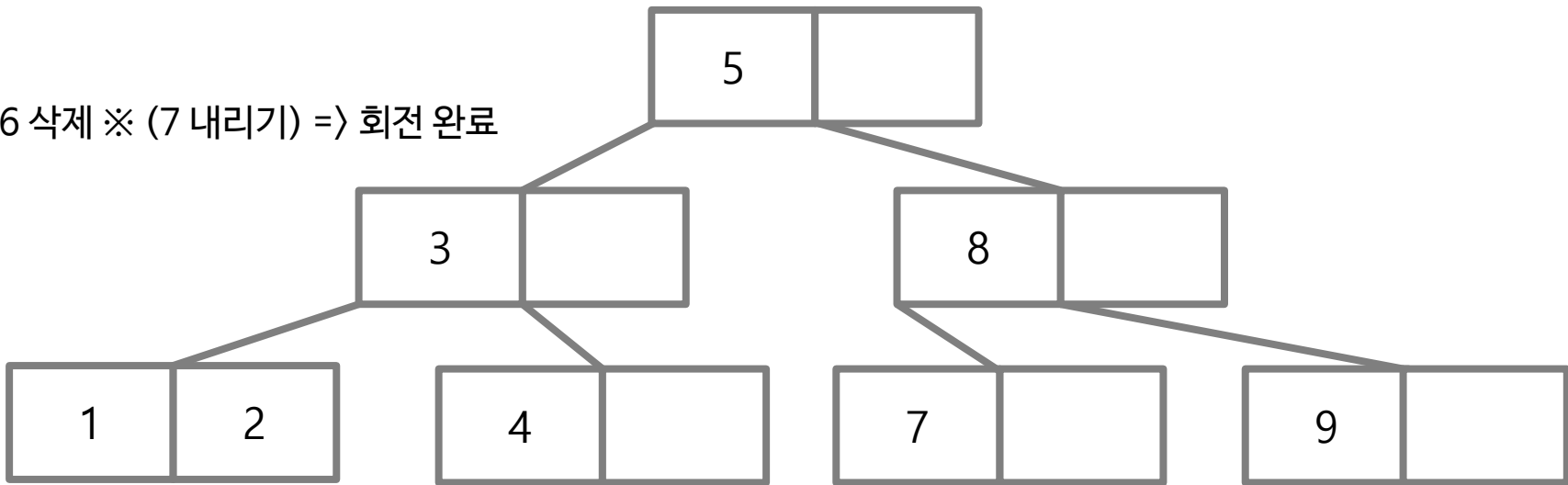
※ 6 삭제 ※ (8 올리기)



B 트리의 개요와 알고리즘 원리

B Tree 삭제 예제 (Degree = 3)

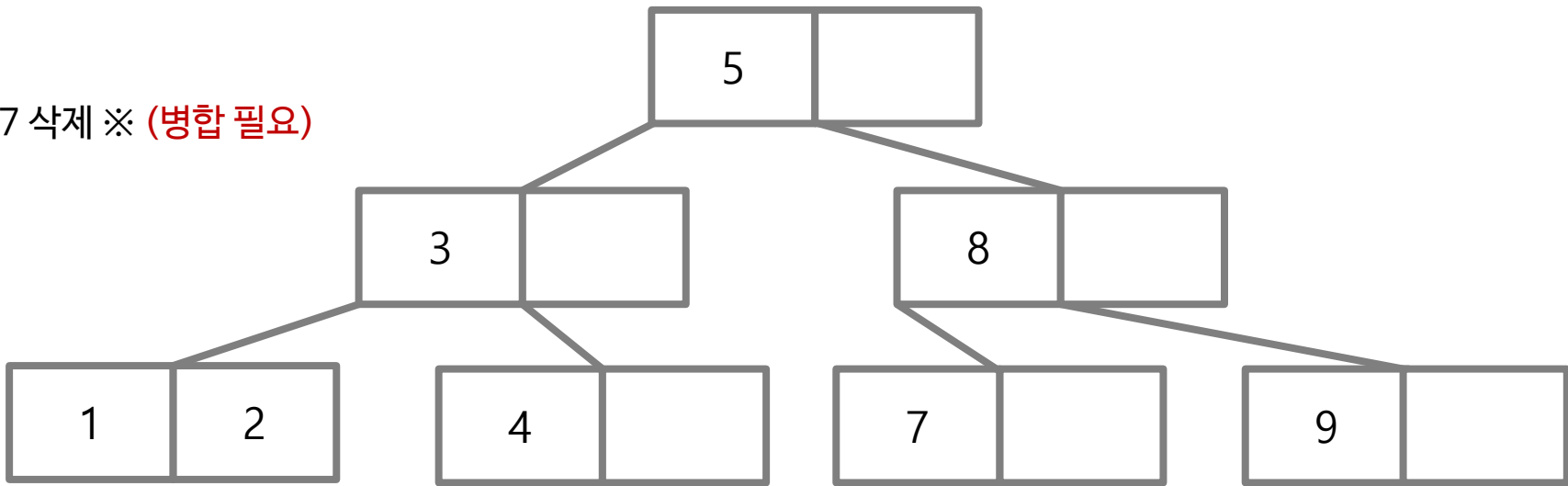
※ 6 삭제 ※ (7 내리기) => 회전 완료



B 트리의 개요와 알고리즘 원리

B Tree 삭제 예제 (Degree = 3)

※ 7 삭제 ※ (병합 필요)

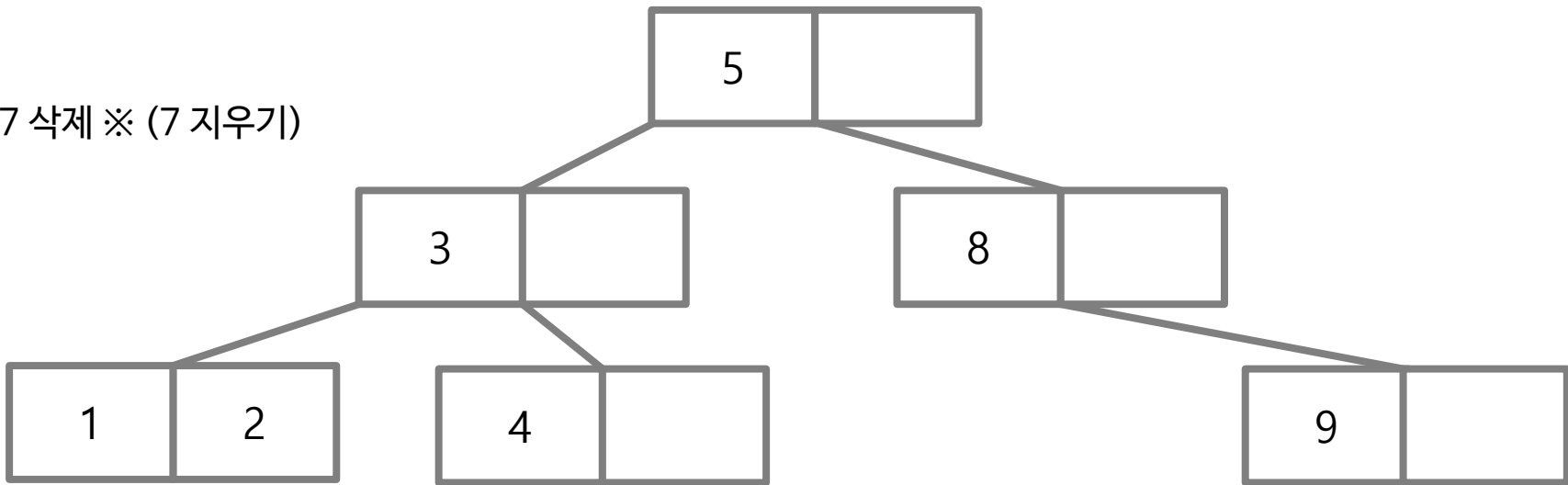


가져올 수 없음

B 트리의 개요와 알고리즘 원리

B Tree 삭제 예제 (Degree = 3)

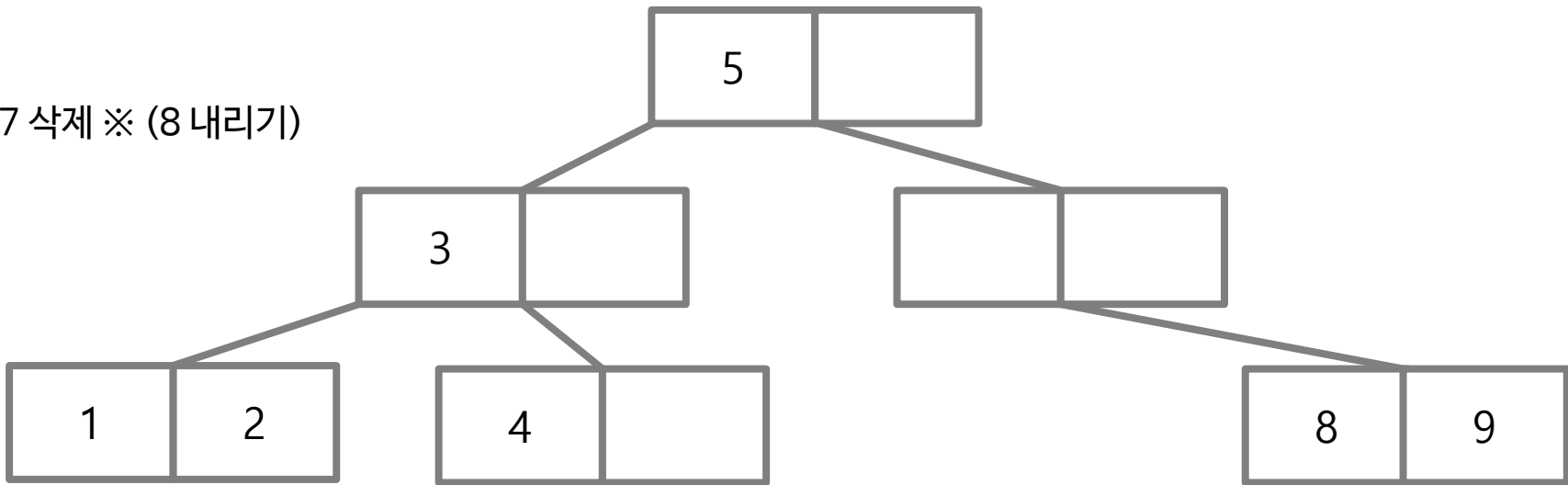
※ 7 삭제 ※ (7 지우기)



B 트리의 개요와 알고리즘 원리

B Tree 삭제 예제 (Degree = 3)

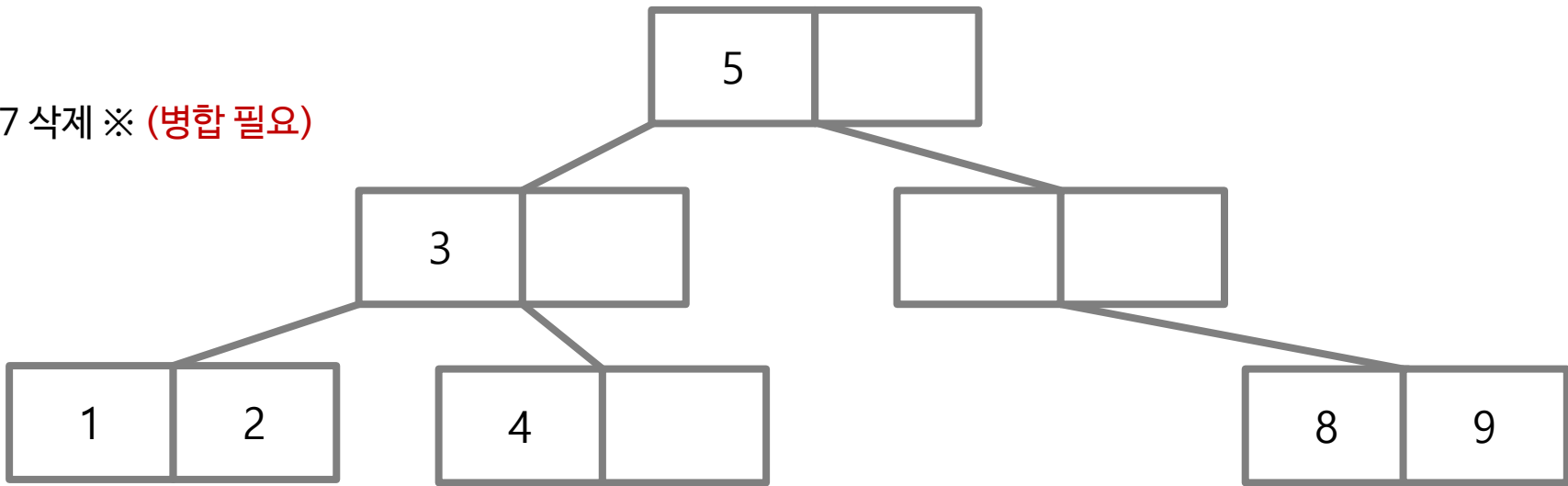
※ 7 삭제 ※ (8 내리기)



B 트리의 개요와 알고리즘 원리

B Tree 삭제 예제 (Degree = 3)

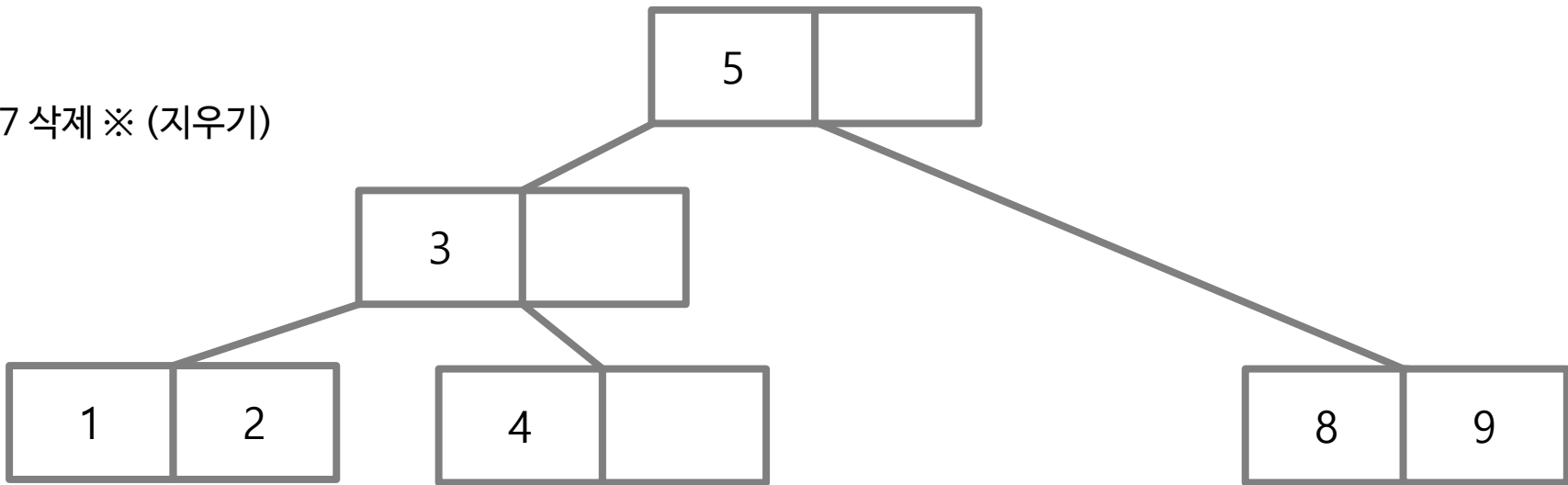
※ 7 삭제 ※ (병합 필요)



B 트리의 개요와 알고리즘 원리

B Tree 삭제 예제 (Degree = 3)

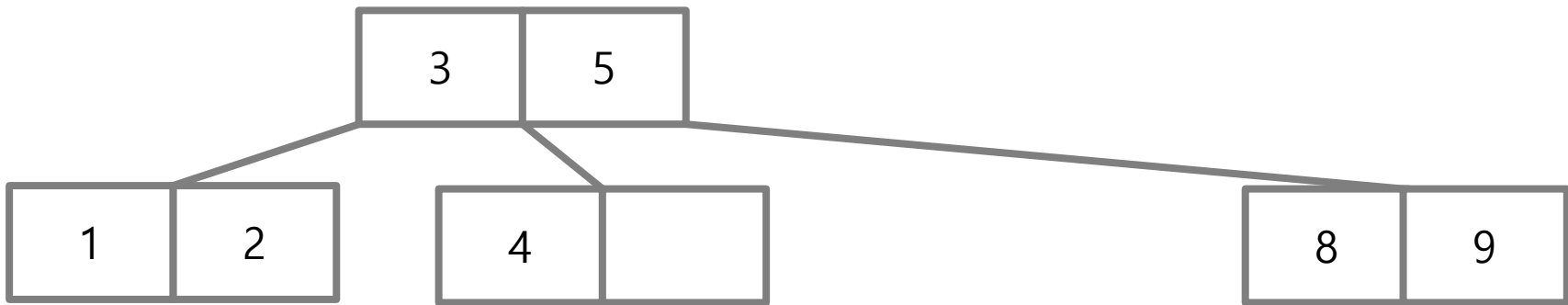
※ 7 삭제 ※ (지우기)



B 트리의 개요와 알고리즘 원리

B Tree 삭제 예제 (Degree = 3)

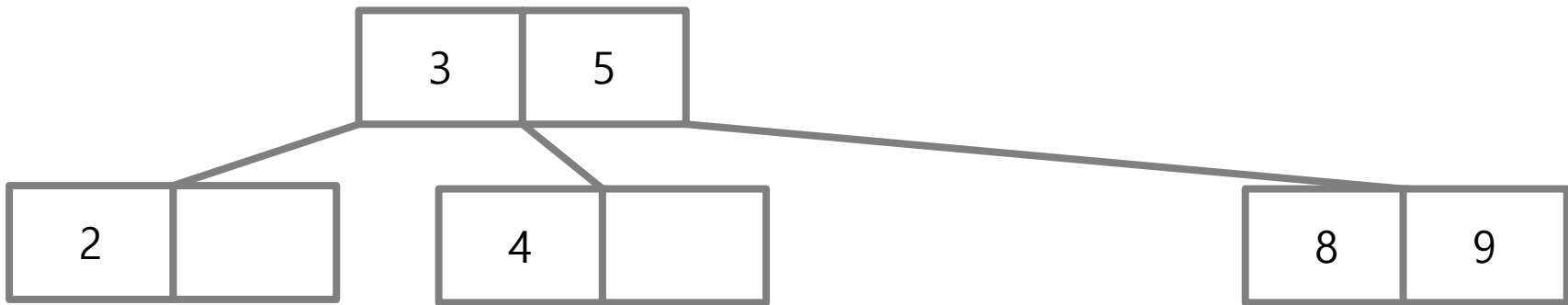
※ 7 삭제 ※ (5 내리기) => 병합 완료



B 트리의 개요와 알고리즘 원리

B Tree 삭제 예제 (Degree = 3)

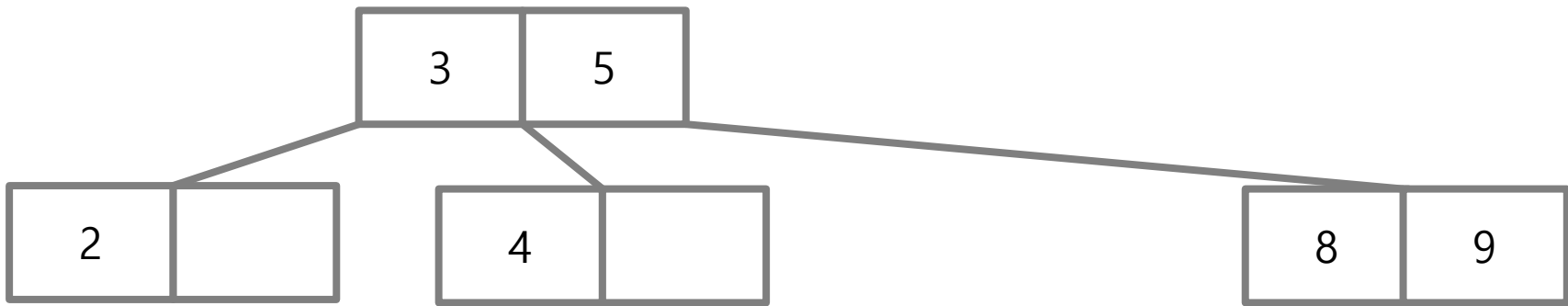
※ 1 삭제 ※



B 트리의 개요와 알고리즘 원리

B Tree 삭제 예제 (Degree = 3)

※ 2 삭제 ※ (병합 필요)

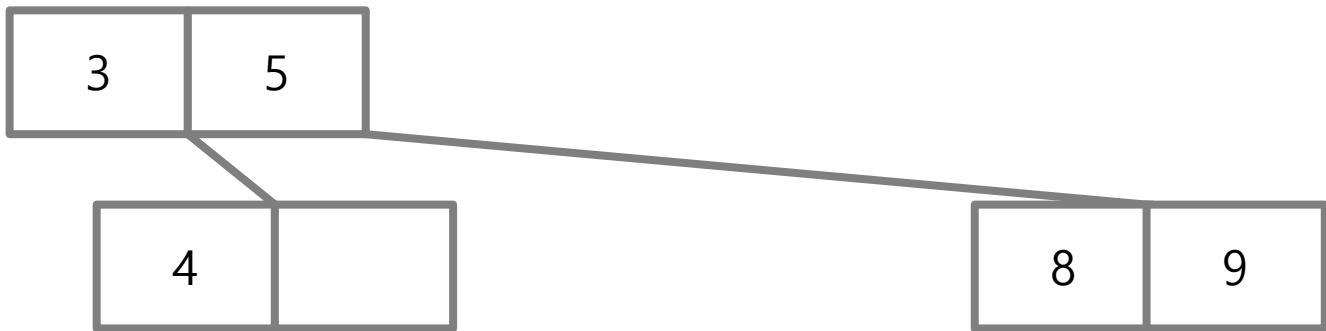


가져올 수 없음

B 트리의 개요와 알고리즘 원리

B Tree 삭제 예제 (Degree = 3)

※ 2 삭제 ※ (2 지우기)



B 트리의 개요와 알고리즘 원리

B Tree 삭제 예제 (Degree = 3)

※ 2 삭제 ※ (3 내리기)

