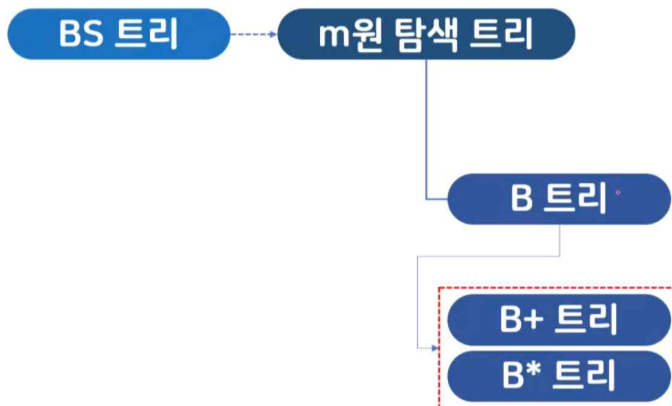


## 제12장 멀티웨이 탐색 트리 I

### 1. m원 탐색 트리



#### ○ 이진 탐색 트리(BS 트리)

- 트리에서 특정 데이터를 검색하고, 노드의 삽입/삭제 연산이 자주 발생하는 응용문제에 가장 효과적인 이진 트리
- ‘왼쪽’과 ‘오른쪽’이라는 방향성을 가지며 다루기가 매우 편리함
- 부모노드를 중심으로 [부모보다 큰 데이터 노드]와 [부모보다 작은 데이터 노드]로 구분됨
- 일반적으로 노드의 개수가 많아지면 트리의 높이가 커지게 됨
- BS 트리가 2원(2-way) 탐색 트리임

#### ○ m원 탐색 트리의 정의

- 트리의 노드가 m개 이하의 가지를 가질 수 있는 탐색 트리  
→ 같은 수의 노드를 갖는 이진트리보다 낮은 높이의 m원 트리
- 이진 탐색 트리의 확장된 형태임
- 탐색 트리의 제한을 따르되 2개 이상 m개 이하의 자식을 가질 수 있음

#### ○ m원 탐색 트리의 제한 조건

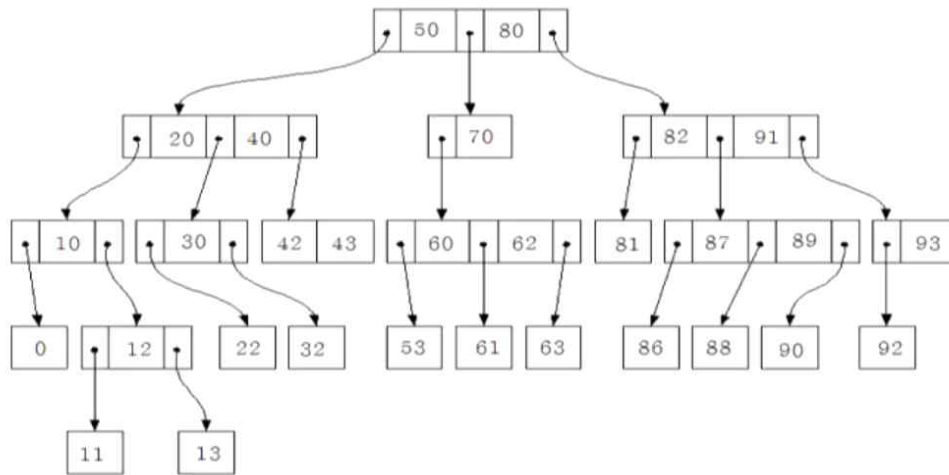
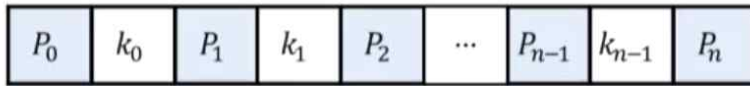
- ① 노드 구조는 다음과 같다. 여기서  $P_0, P_1, \dots, P_n$  은 서브트리에 대한 포인터이고  $k_0, k_1, \dots, k_{n-1}$  은 키 값이다. 또한  $n \leq m-1$  이 성립한다.

$P_0$	$k_0$	$P_1$	$k_1$	$P_2$	$\dots$	$P_{n-1}$	$k_{n-1}$	$P_n$
-------	-------	-------	-------	-------	---------	-----------	-----------	-------

- ②  $i=0, 1, \dots, n-2$  인  $i$ 에 대해  $k_i < k_{i+1}$  을 만족한다.
- ③  $i=0, 1, \dots, n-1$  인  $i$ 에 대해  $P_i$ 가 가리키는 서브트리의 모든 키값은  $k_i$ 의 키값보다 작다.
- ④  $P_n$ 이 가리키는 서브트리의 모든 키값은  $k_{n-1}$ 의 키값보다 크다.
- ⑤  $i=0, 1, \dots, n$  인  $i$ 에 대해  $P_i$ 가 가리키는 서브트리는 m원 탐색 트리이다.

○ m원 탐색 트리 - 3원 탐색 트리

- 잎노드는 키값으로만 표기하고 내부노드 중 자식이 없는 포인터들은 표시하지 않음



○ m원 탐색 트리의 탐색

- 일반적으로 노드의 가지 개수가 많을수록(서브트리가 많을수록), 최대 탐색길이는 짧아짐(트리의 깊이가 얕으므로 더 빨리 찾을 수 있음)

## 2. B 트리

○ B 트리

- m원 탐색 트리는 서브트리의 균형에 대해서는 특별히 제한하지 않음
- 각 노드가 자식을 많이 갖게 하여 트리의 높이를 줄이고 전체적으로 균형을 유지한다면 탐색 성능을 더욱 향상할 수 있음
- m원 탐색트리를 개선한 B트리는 인덱스 구조를 구현하는데 가장 일반적으로 사용함
- 차수 m인 B 트리의 탐색 경로 길이는 같은 개수의 키를 가지는 이상적인 m원 탐색 트리보다 길 수 있지만, 키값을 삽입/삭제할 때 B 트리를 유지하는 것이 더 쉬움

○ B 트리의 조건

- ① 루트와 잎노드를 제외한 트리의 각 노드는 최소  $\lceil m/2 \rceil$  개의 서브트리를 갖는다.
  - ※  $\lceil \cdot \rceil$  - 가우스 함수: 올림
- ② 트리의 루트는 최소한 2개의 서브트리를 갖는다.
- ③ 트리의 모든 잎노드는 같은 레벨에 있다.

### 3. B\*, B+ 트리

#### ○ B\* 트리의 정의

- 노드의 약  $2/3$  이상이 채워지는 B 트리
- 노드가 꽉 차면 분리하지 않고, 키와 포인터를 재배치하여 다른 형제 노드로 옮김
- 삽입/삭제 시 발생하는 노드 분리를 줄이려고 고안됨

#### ◆ 차수가 m인 B\* 트리는 다음 조건을 만족하는 B 트리이다

- ① 공집합이거나 높이가 1 이상인 m원 탐색 트리이다.
- ② 루트 노드는 2개 이상  $2\lfloor (2m-2)/3 \rfloor + 1$ 개 이하의 자식노드를 갖는다.
- ③ 내부노드는 최소한  $\lceil (2m-1)/3 \rceil$ 개의 자식노드를 갖는다.
- ④ 모든 잎노드는 동일한 레벨에 놓인다.
- ⑤ 포인터가 k개인 잎이 아닌 노드는 k-1 개의 키를 갖는다. (루트노드 포함)

#### ○ B+ 트리의 정의

- 탐색 트리로 구성하면 매우 빠르게 탐색할 수 있지만, 전체 데이터를 차례로 처리하기는 불편함
  - 매번 왼쪽인지 오른쪽인지 비교해가면서 다음 노드를 찾아가면서 처리해야 함
- B+ 트리는 인덱스된 순차 파일을 구성하는데 사용됨
- B 트리와 같이 각 노드의 키가 적어도  $1/2$ 이 채워져야 하는 점은 같음
- 잎노드를 순차적으로 연결하는 포인터 집합이 있다는 점에서 다름
  - 잎노드의 마지막 포인터는 다음 키값을 갖는 노드를 가리킴
  - 순차 처리를 할 때는 이 포인터를 이용해서 (키값을 비교하지 않고) 차례로 다음 데이터에 접근해서 처리
- 모든 키값이 잎노드에 있고 그 키값에 대응하는 실제 데이터 (파일 내용)에 대한 주소를 잎노드만이 가지고 있음
  - 직접 탐색은 잎노드에 도달해야 종료