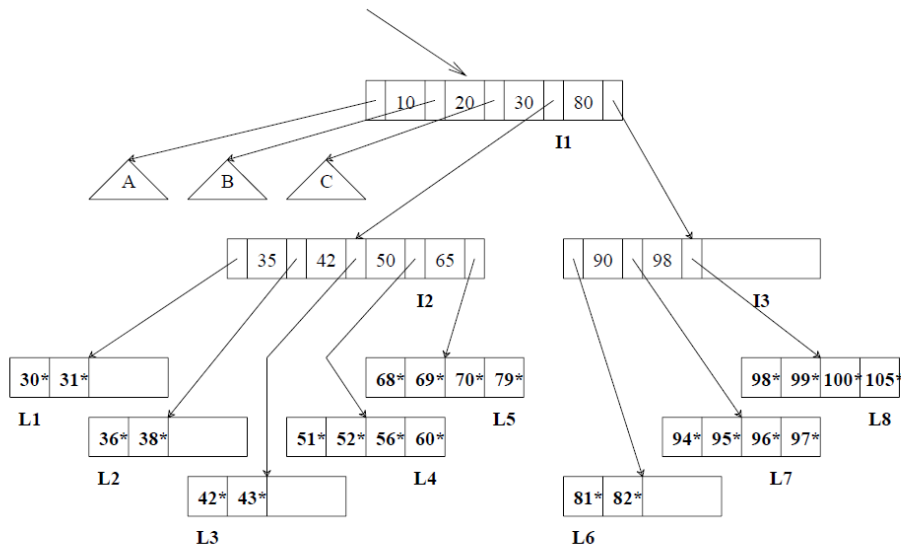


고급 데이터베이스

B+Tree

이름(학번)	황 준 일 (32131766)
담당교수	이 석 균 교수님
제출일	2019. 11. 21



Consider the B+ tree index shown in Figure above, which uses Alternative (1) for data entries. Each intermediate node can hold up to five pointers and four key values. Each leaf can hold up to four records, and leaf nodes are doubly linked as usual, although these links are not shown in the figure. Answer the following questions.

1. Name all the tree nodes that must be fetched to answer the following query: "Get all records with search key greater than 38."

I1 → I2 → L2 → L3 → L4 → L5 → I3 → L6 → L7 → L8

2. Show the B+ tree that would result from inserting a record with search key 109 into the tree.

I3 → [L6] 90 [L7] 98 [L8] 100 [L9]

L8 → 98 | 99

L9 → 100 | 105 | 109

3. Show the B+ tree that would result from deleting the record with search key 81 from the original tree.

L6 → 82 | 94

L7 → 95 | 96 | 97

4. Name a search key value such that inserting it into the (original) tree would cause an increase in the height of the tree.

X = 65...79

L5 분할 불가(L4가 꽉참) → I2분할 → I1분할

결과적으로 depth가 증가합니다.

5. Note that subtrees A, B, and C are not fully specified. Nonetheless, what can you infer about the contents and the shape of these trees?

A에는 0~10 사이의 노드들이 있을 것이며

B에는 10~20, C에는 20~30 사이의 노드들이 있을 것이다.

그리고 A, B, C는 각각 2개 이상의 key와 3개 이상의 pointer가 있다.

이러한 것을 유추할 수 있는 이유는 Root Node를 알고 있으며, B+tree의 생성규칙을 알고 있기 때문이다.

6. How would your answers to the preceding questions change if this were an ISAM index?

sol)

- 1) 검색은 동일
- 2) L8에서부터 시작하는 overflow linked list 를 만들어야 함.
- 3) 81이 해당 위치에서 삭제되고 끝. 분할(혹은 이동) 과정이 생략됨
- 4) 트리의 구조를 변경할 수 없음
- 5) A, B, C 각각의 높이는 12, 13의 높이와 똑같은 것이다. 다만 A, B, C와 그 자식 node의 최소 개수는 알 수 없다.

7. Suppose that this is an ISAM index. What is the minimum number of insertions needed to create a chain of three overflow pages?

삽입은 L4, L5, L7, L8 중 하나에 이루어질 것이며, overflow page는 최대 2개의 노드를 갖는다. 따라서 최소 5번의 삽입이 필요하다.

1,2번째 삽입 → 1번째 page

3,4번째 삽입 → 2번째 page

5번째 삽입 → 3번째 page