Design and Analysis of Algorithms: Exercise 18.2-5

Ben Chaplin

Exercise 18.2-5 from *Introduction to Algorithms* by Cormen, Leiserson, Rivest & Stein (3rd ed)

Contents

1 Problem statement

In B-trees, since leaf nodes require no pointers to children, they could conceivably use a different (larger) t value than internal nodes for the same disk page size. Show how to modify the procedures for creating and inserting into a B-tree to handle this variation

2 Improved B-tree

2.1 Data structure

Given t, we can assume our disk page size had room for $2t$ pointers to children. Let p be the size of a pointer, k be the size of a key, and c be a constant such that $cp = k$. Then we can support $pt = \frac{k}{c}t$ new keys.

2.2 B-tree creation

Algorithm 1 Original B-tree create

Input
T pointer to the tree

1: $x =$ ALLOCATE-NODE()
2: $x.leaf =$ **True**
3: $x.n = 0$
4: DISK-WRITE(x)
5: $T.root = x$

Nothing needs to be changed to support the improved B-tree.

2.3 B-tree insertion

(No changes required to B-TREE-SPLIT-CHILD)

Algorithm 2 B-TREE-INSERT

Input
T pointer to the tree
k key to insert

```
1: r = T.root
2: if r.n == 2t − 1 or (r.leaf and r.n == (2t − 1) + pt) then
3:     s = ALLOCATE-NODE()
4:     T.root = s
5:     s.leaf = False
6:     s.n = 0
7:     s.c_1 = r
8:     B-TREE-SPLIT-CHILD(s, 1)
9:     B-TREE-INSERT-NONFULL(s, k)
10: else
11:     B-TREE-INSERT-NONFULL(r, k)
12: end if
```

Algorithm 3 B-TREE-INSERT-NONFULL

Input
x node under which to insert
k key to insert

```
1: i = x.n
2: if x.leaf then
3:     while i ≥ 1 and k < x.key_i do
4:         x.key_{i+1} = x.key_i
5:         i = i − 1
6:     end while
7:     x.key_{i+1} = k
8:     x.n = x.n + 1
9:     DISK-WRITE(x)
10: else
11:     while i ≥ 1 and k < x.key_i do
12:         i = i − 1
13:     end while
14:     i = i + 1
15:     DISK-READ(x.c_i)
16:     if x.c_i.n == 2t − 1 or (x.c_i.leaf and x.c_i.n == (2t − 1) + pt then
17:         B-TREE-SPLIT-CHILD(x, i)
18:         if k > x.key_i then
19:             i = i + 1
20:         end if
21:     end if
22:     B-TREE-INSERT-NONFULL(x.c_i, k)
23: end if
```