

BtreeInsertion(T, k)

$r \leftarrow \text{root}[T]$

if $n[r] = 2t - 1$

$S \leftarrow \text{AllocateNode}()$

$\text{root}[T] \leftarrow S$

$\text{leaf}[S] \leftarrow \text{FALSE}$

$n[S] \leftarrow 0$

$c[S] \leftarrow r$

BtreeSplitChild(C, i, r)

BtreeInsertNonFull(C, k)

else BtreeInsertNonFull(C, k)

$i \leftarrow n[\alpha]$

if $\text{leaf}[\alpha]$

while $i \geq 1$ and $k < \text{key}[i][\alpha]$:

$\text{key}[i+1][\alpha] \geq \text{key}[i][\alpha]$

$i \leftarrow i - 1$

$\text{key}[i+1][\alpha] \geq k$

$n[\alpha] \leftarrow n[\alpha] + 1$

else while $i \geq 1$ and $\text{key}[i][\alpha] < k$

$i \leftarrow i - 1$

$i \leftarrow i + 1$

if $n[c[i][\alpha]] \geq 2t - 1$

BtreeSplitChild($C, i, c[i][\alpha]$)

if $k \leq \text{rt}$

$\text{key}[i][\alpha]$

$i \leftarrow i + 1$

BtreeInsertNonFull($C[i][\alpha], k$)

Hansh Shankar

IBM18CS032

BtreeSplitChild(x, i)

BtreeSplitChild(x, i, y)

z = AllocateNode()

key[z] = key[y]

n[z] = t-1

for j = 1 to t-1

key[j][z] = key[j+t][y]

if not key[y]

for j = 1 to t

c[j][z] = c[j+t][y]

n[y] = t-1

for j = n[x]+1 to i+1

c[j+1][x] = c[j][x]

c[i+1][x] = 0

for j = n[x] to i

key[j+1][x] = key[j][x]

key[i][x] = key[t][y]

n[x] = n[x] + 1

BtreeSplitChild

BtreeSplitChild(y, i)

z = AllocateNode()

n[z] = ~~t~~ t[y] - 1

for $j = 0; j < t-1; j++$

~~$key[j] = key[j+1]$~~

$key[j] = key[j+t]$

if $del[y] == FALSE$

for $j = 0$ to $t-1$

$c[j] = c[j+t]$

~~$n[y] = t-1$~~

for $j = n$ to $i+1$

$c[j] = c[j+1]$

$c[i+1] = 0$

for $j = n-1$ to i

$key[j+1] = key[j]$

$key[i] = key[i+y]$

$n = n+1$