```
B-Tree-Delete-Key(x, k)
    if not leaf[x] then
        y ← Preceding-Child(x)
        z ← Successor-Child(x)
        if n[y]>t−1then
            k' ← Find-Predecessor-Key(k, x)
            Move-Key(k', y, x)
            Move-Key(k, x, z)
            B-Tree-Delete-Key(k, z)
        else if n[z] > t − 1 then
            k' ← Find-Successor-Key(k, x)
            Move-Key(k', z, x)
            Move-Key(k, x, y)
            B-Tree-Delete-Key(k, y)
        else
            Move-Key(k, x, y)
            Merge-Nodes(y, z)
            B-Tree-Delete-Key(k, y)
else (leaf node)
        y ← Preceding-Child(x)
        z ← Successor-Child(x)
        w ← root(x)
        v ← RootKey(x)
        if n[x] > t − 1 then Remove-Key(k, x)
        else if n[y] > t − 1 then
            k' ← Find-Predecessor-Key(w, v)
            Move-Key(k', y, w)
            k' ← Find-Successor-Key(w, v)
            Move-Key(k', w, x)
            B-Tree-Delete-Key(k, x)
        else if n[w] > t − 1 then
            k' ← Find-Successor-Key(w, v)
            Move-Key(k', z, w)
            k' ← Find-Predecessor-Key(w, v)
```

```
                    Move-Key(k', w, x)
                    B-Tree-Delete-Key(k, x)
else
s ← Find-Sibling(w)
w' ← root(w)
        if n[w′]=t−1 then
                    Merge-Nodes(w′, w)
                    Merge-Nodes(w, s)
                    B-Tree-Delete-Key(k, x)
        else
                    Move-Key(v, w, x)
                    B-Tree-Delete-Key(k, x)
```