

Oznaczenia:

T - stopień drzewa

x - węzeł B-drzewa

$x.n$ - ilość kluczy w węźle x

$x.leaf$ - czy x jest liściem

$x.k[i]$ - klucze w węźle x ; $i=1,2,\dots,(x.n)$

$x.c[i]$ - synowie węzła x ; $i=1,2,\dots,(x.n+1)$

```

B-Tree-Delete(x,k)
// x.n>T-1 lub x jest korzeniem
if x zawiera k
  if x.leaf
    usuń k z x
    zapisz x na dysk
    // ta operacja może opróżnić korzeń i wtedy
    // drzewo zostaje puste!
  else // x nie jest liściem, x zawiera k
    wyszukujemy i takie, że k == x.k[i]
    przeczytaj x.c[i] oraz x.c[i+1]
    przypisz: l = x.c[i], p = x.c[i+1]
    if l.n>T-1
      x.k[i] = B-Tree-Delete-Max(l)
    else if p.n>T-1
      x.k[i] = B-Tree-Delete-Min(p)
    else B-Tree-Join(x,i,l,p)
      B-Tree-Delete(x.c[i],k)
  else // x nie zawiera k, x nie jest liściem
    wyznacz i takie, że klucz k powinien być w x.c[i]
    // czyli k jest pomiędzy x.k[i-1] a x.k[i]
    przeczytaj x.c[i-1], x.c[i], x.c[i+1]
    // może się zdarzyć, że istnieją tylko dwa z nich
    przypisz: l = x.c[i-1], s = x.c[i], p = x.c[i+1]
    if s.n>T-1
      B-Tree-Delete(s,k)
    else if i>1 (czyli l istnieje) oraz l.n>T-1
      B-Tree-Right-Rotate(l,x,s)
      B-Tree-Delete(s,k)
    else if i<x.n+1 (czyli p istnieje) oraz p.n>T-1
      B-Tree-Left-Rotate(s,x,p)
      B-Tree-Delete(s,k)
    else if i>1
      B-Tree-Join(x,i-1,l,s)
      B-Tree-Delete(l,k)
    else B-Tree-Join(x,i,s,p)
      B-Tree-Delete(s,k)

```

```

B-Tree-Delete-Min(x)
// usuwa minimalny klucz w poddrzewie o korzeniu x,
// czyli skrajny lewy w skrajnym lewym liściu;
// wynikiem jest usunięty klucz
    if x.leaf
        k1=x.k[1]
        usuń z "x" klucz x.k[1] przesuwając inne klucze w prawo
        zapisz na dysk x
        return k1
    else
        // rekursywne zejście do x.c[1]
        przeczytaj x.c[1], x.c[2]
        przypisz l = x.c[1], s = x.c[2]
        if l.n>T-1
            return B-Tree-Delete-Min(l)
        else if s.n>T-1
            B-Tree-Left-Rotate(l,x,s)
            B-Tree-Delete-Min(l)
        else B-Tree-Join(x,l,s)
            B-Tree-Delete-Min(l)

```

```

B-Tree-Delete-Max(x)
    analogicznie jak B-Tree-Delete-Min(x)

```

```

B-Tree-Join(x,i,y,z)
    łączy węzły y oraz z, gdzie y==x.c[i], z==x.c[i+1]
    zapisuje zmienione węzły na dysk

```

```

B-Tree-Left-Rotate(x,y,z)
    rotacja w lewo; x - ojciec, y -lewy syn, z - prawy syn
    zapisuje zmienione węzły na dysk

```

```

B-Tree-Right-Rotate(x,y,z)
    rotacja w prawo; x - ojciec, y -lewy syn, z -prawy syn
    zapisuje zmienione węzły na dysk

```

