```
void   decreaseKeyVal( old_val, new_val)
{
    BinomialHeapNode  temp = Node . findNode (old_val)

    if ( temp == null )
        return

    temp . key = new_val;
    BinomialHeapNode  tempParent = temp . parent;

    while (( tempParent != null ) && (temp. key < tempParent . key ))
    {
        int   z = temp . key ;
        temp . key = tempParent . key
        tempParent . key = z;

        temp = tempParent;
        tempParent = tempParent . parent;
    }
}
```

```
void delete (int val)
{
    if (Node != null && Node.findKey (val) != null)
    {
        decreaseKeyVal (val, findMin () - 1)
        extractMin ();
    }
}
```