Delete (H) & Decrease Key (H) function on Binomial Heap

// Decrease key (H) Function

```
Void decreasekey BHeap (Node *H, int old_val, int new_val)
{
    Node * node = findNode (H, old_val);
    if (node == NULL)
        return;
    node -> val = new_val;
    Node * parent = node -> parent;
    while (parent != NULL && node -> val < parent -> val)
    {
        swap (node -> val, parent -> val);
        node = parent;
        parent = parent -> parent;
    }
}
```

// function to delet an element

```
Node * binomial Heap Delete (Node *h, int val)
{
    if (h == NULL)
        return NULL;
    decreasekey BHeap (h, val, INT_MIN);
    return extractMinBheap (h);
}
```


clf.

// Function findNode

```
Node * findNode (Node * h, int val)
{
  if (h==NULL)
      return;
  if (h->val == val)
      return h;
  Node *res = findNode (h->child, val);
      if (res != NULL)
          return res;
      return findNode (h->Sibling, val);
}
```


Chf.