# Binomial Heap Operations

Sahana L
1BM18CS089

```
// DecreaseKeyBHeap
void decreaseKeyBHeap(Node *H, int old_val, int new_val)
{
    //Check element is present or not
    Node *node = findNode(H, old_val);

    // return if node is not present
    if (node == NULL)
        return;

    // reduce the value to minimum
    node->val = new_val;
    Node *parent = node->parent;

    //Update the heap according to reduced value
    while( parent != NULL && node->val < parent->val)
    {
        swap (node->val, parent->val);
        node = parent;
        parent = parent->parent;
    }
}

// Delete
Node *binomialHeapDelete (Node *h, int val)
{
    // Check if heap is empty or not
    if (h == NULL)
        return NULL;

    // Reduce the value of element to minimum
    decreaseKeyBHeap(h, val, INT_MIN);

    // Delete min element from HEAP
    return extractMinBHeap(h);
}
```