Raghav
Mittal
18N/8CS075

# Binomial Heap → Delete & Decrease

## Decrease.

Decrease key is compared with its parent and if parent key is more, we swap keys and recur for parent. We stop when we either reach a node where parent has smaller key ~~or~~

```
fn : void decreaseKeyBHeap (Node * x, int
                            old_val, int
                            new_value)
Node * node = findNode (x, old_val);
    if (node == NULL)
        return;

node → val = new_val;
Node * parent = node → parent;

while ( parent→val > node → val)
{
    swap (node → val, parent → val);
    node = parent;
    parent = parent → parent;
}
};
```

Deepe

first reduce the key to minus infinite and then call extractMin() function.

Eg:

```
if (h == NULL)
    return NULL;
decreaseKey(h, val, INT_MIN);
return extractMinHeap(h);
```

3.