

```
void decreaseKey BHeap ( Node *H, int old,  
int new);
```

```
{ Node * node = findNode (H, old);  
if (node == NULL)  
return;
```

```
node->val = new;
```

```
Node * parent = node->parent;
```

```
}
```

```
while (parent && parent->val > node->val)  
{
```

```
    swap (node->val, parent->val);
```

```
    node = parent;
```

```
    parent = parent->parent;
```

```
}
```

```
}
```

```
Node * heapDelete (Node *v, int val) {  
    if (!v)  
        return null;
```

```
    decreaseKey BHeap (v, val, INT_MIN);  
    return extract Min BHeap (v);
```

```

struct Node {
    int val, degree;
    Node *parent, *child, *sibling;
    int key;
    Node()
    {
        val = key;
        degree = 0;
        child = sibling = parent = NULL;
    }
};

```