

16-12-20

Sanjeew Kumar Singh  
1BM18CS093  
ADS lab 10 - Prgm 10

classmate

Date

Page

Pg 1

Prgm 10 - lab 11

# function to decrease the value

```
void DecreaseKeyBHeap(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 delete an element

```
Node *binomialHeapDelete(Node *h, int val) {
```

```
    if (h == NULL)
```

```
        return NULL;
```

```
    decreaseKeyBHeap(h, val, int-min);
```

```
    return extractMinHeap(h);
```

```
}
```

Sanjeev Kumar Singh  
12M18CS093  
lab Prgm 10

classmate

Date

Page

pg 2

16-12-20

```
# extract min value
Node * extractMinBHeap (Node * h) {
    if (h == NULL)
        return NULL;

    Node * minNode = h;
    int minVal = h->val;
    Node * curr = h;

    while (curr->Sibling != NULL) {
        if ((curr->Sibling)->val < minVal) {
            minNode = (curr->Sibling);
            minNode->prev = curr;
            curr = curr->Sibling;
        }
    }

    if (minNode->prev == NULL && minNode->Sibling == NULL)
        h = minNode;
    else if (minNode->prev == NULL)
        h = minNode->Sibling;
    else
        minNode->prev->Sibling = minNode->Sibling;

    if (minNode->Child != NULL) {
        reverseList (minNode->Child);
        (minNode->Child)->Sibling = NULL;
    }

    return UnionBHeaps (h, root);
}
```