Shandan A.M.
1BM18CS025

DATE | | PAGE:

10) Delete and decreaseKey :

(a) delete (H) :

```
Node * binomialHeapDelete (Node *h , int val) {
        if (H == NULL)
                return NULL
        decreaseKey (h, val, INT_MIN);
        return ExtractMinBHeap (h);
}
```

(b) decreaseKey (H) :

```
void decreaseKey (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;
        }
}
```

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