16/12/2020

ADS

KARTHIK VENUGOPAL

1BM18CS043

BINOMIAL HEAP WRITEUP

```
function delete(Node *h, int val)
{
    if (!h) return NULL;
    decreaseKeyBHeap(h, val, INT_MIN);
    return extractMinHeap(h);
}
function decreaseKeyBHeap(Node* H, int oldv, int newv)
{
    Node* node = findNode(H, oldv);
    if (!node) return;
    node->val = newv;
    Node * parent = node->parent;
    while (parent != NULL && node->val < parent->val)
    {
        swap(node->val, parent->val);
        node = parent;
        parent = parent->parent;
    }
}
```

APS KARTHIK

```
Function * extractMinHeap (Node *h) {
    if (!h) return NULL;
    Node * min_prev = NULL;
    Node * min = h;
    int min = h -> val;
    node * curr = h;
    while (curr -> sibling != NULL) {
        if ((curr -> sibling) -> val < min) {
            min = curr -> sibling -> val;
            min_prev = curr;
            min = curr -> sibling;
        }
        curr = curr -> sibling;
    }
    if (min_prev == NULL && min -> sibling == NULL) h = NULL;
    else if (min_prev == NULL) h = min -> sibling;
    else min_prev -> sibling = min -> sibling;
    if (min -> child) {
        reverselist (min -> child);
        min -> child -> sibling = NULL;
    }
    return unionBHeap (h, root);
}
```

APS

KARTHIK

```
function findNode(Node* h, int val){

    if(!h) return NULL;
    if(h->val == val) return h;
    Node* res = findNode(h->child, val);
      if(res!= NULL) return res;
      return findNode(h->sibling, val);

}
function revslist(Node* h){

      if(h->sibling){

        revslist(h->sibling);
        h->sibling->sibling=h;

      }
      else
        root=h;

}
```