

3

4

3

2

4

```

    if (min->prev == NULL && min->sibling == NULL) h->prev = min;
    else if (min->prev == NULL) h->prev = min->sibling;
    else min->prev->sibling = min->sibling;
    if (min->child) {
        reverseList(min->child);
        min->child->sibling = NULL;
    }

```

```

    return unionBheap(h, root);
}

```

```

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);
}

```

```

reverseList (Node *h) {
    if (h->sibling) {
        reverseList(h->sibling);
        h->sibling->sibling = h;
    } else root = h;
}

```