

Siddhartha Lab - 16/12/20

Struct Node {

int val, degree;

Node \* parent, \* child, \* sibling;

int key

Node() {

val = key;

degree = 0;

child = sibling = parent = NULL;

}

};

void decreaseKey BHeap (Node \*H, int old, int new) {

Node \* node = findNode(H, old);

if (node == NULL)

return;

node -&gt; val = new &lt; val;

Node \* parent = node -&gt; parent;

while (parent != NULL &amp;&amp; parent -&gt; val &gt; node -&gt; val) {

swap (node -&gt; val, parent -&gt; val);

node = parent;

parent = parent -&gt; parent;

}

}

Node \* heapDelete (Node \*h, int val) {

if ( /n)

return NULL;

decreaseKey BHeap (h, val, INT\_MIN);

return extractMinBHeap(h);

}