void downHeap(Node \*n) {

Node \* swap\_with = nullptr;

// Implement downHeap() here.

// if no child, do nothing

if(!n->left && !n->right){}

else if(!n->left && n->right){

// check if value is greater than the right child

if(n->value > n->right->value){

swap\_with = n->right;

}

}else if(n->left && !n->right){

// check if value is greater than the left child

if(n->value > n->left->value){

swap\_with = n->left;

}

}else{

// both children exist

if(n->value <= n->left->value && n->value <= n->right->value){}

else{

if(n->left->value < n->right->value){

swap\_with = n->left;

}else{

swap\_with = n->right;

}

}

}

if(swap\_with){

int temp = n->value;

n->value = swap\_with->value;

swap\_with->value = temp;

downHeap(swap\_with);

}

// Implement downHeap() here.

}