Jameson Connor Harrington

Professor Rahul Shah

CSC 3102

26 June 2020

Homework 2

1. First the array must be used to build heap which takes O(n) time by performing minHeapify on the array from n/2-1 to 0. Then just extract min in a loop that runs k times. This loops runs in O(klogn) time. Together it runs in O(n+klogn) time.
2. Range Queries:

```
Int RangeCount(k1,k2,currentNode){
        If(currentNode==NULL)
                Return 0;
        Int count=0;
        If(k1<currentNode->data<k2){
                count++;
                count+=RangeCount(k1,k2,currentNode->left);
                count+=RangeCount(k1,k2,currentNode->right);
                return count;
        else if(currentNode->data < k1)
                return RangeCount(k1,k2,currentNode->right)
        else return RangeCount(k1,k2,currentNode->left)
}
Void RangeReport(k1,k2,currentNode){
        If(currentNode==NULL)
```

```
        Return;
    Else If(k1<currentNode->data<k2){
        print(currentNode->data);
        RangeReport(k1,k2,currentNode->left);
        RangeReport(k1,k2,currentNode->right);
        return count;
    else if(currentNode->data < k1)
        return RangeReport(k1,k2,currentNode->right)
    else return RangeReport(k1,k2,currentNode->left)
}
```

Add the mindata field to Node that has the value of the lowest data in all of its children or itself. If a node is added to the bottom of a tree or deleted then the subsequent parent nodes will be updated if it conflicts.

```
Node* RangeMin(k1,k2,currentNode,minData){
    If(k1<currentNode->data<k2 & currentNode->mindata<minData){
        tempNode1= RangeReport(k1,k2,currentNode->left);
        tempNode2= RangeReport(k1,k2,currentNode->right);
        If(tempNode1==NULL | tempNode2==NULL)
            Return currentNode;
        If(tempNode1!=NULL | tempNode2==NULL)
            Return tempNode1;
        If(tempNode1==NULL | tempNode2!=NULL)
            Return tempNode2;
```

```
        else if(currentNode->data < k1)
                return RangeReport(k1,k2,currentNode->right)
        else return RangeReport(k1,k2,currentNode->left)
}
```

3. Pseudocode:

```
current_node=root;
While(current_node != NULL){
        print(current_node->data);
        Q.insert_last(current_node->left);
        Q.insert_last(current_node->right);
        if(q.is_empty()){
                current_node=NULL;
        else {
                current_node=Q.delete_first();
        }
}
```