

Renat Norderhaug

CS 477

Homework 5

1. We want it to decrease rather than increase because then we won't be allowed to call MAX-HEAPIFY, since it will fail the condition of having the subtrees be max-heaps. If we start with 1, then there is no guarantee that A[2] and A[3] are roots of max-heaps.
2. if in line 10, the loop goes from 1 to A.length

```
for (j = 1; j <= A.length; j++)
```

```
{
```

```
  b[c[a[j]]] = a[j];
```

```
  c[a[j]] = c[a[j]] - 1;
```

```
}
```

This is still correct and will still give the sorted output, because the correctness of the algorithm does not depend upon the order in which A is processed either from A.length to 1 or 1 to A.length. In both cases we get B as a sorted array. Our C array is well established. so we get sorted output.

However, the modified algorithm is not stable because the element taken from later started out with a higher index than one taken earlier.

- 3.

```
hw5.cpp
~/Desktop/CS477/hw5
Save

Open +
/*
Author: Renat Norderhaug
Class: CS 477
Project: heap check in C++
Date: 10/10/19
*/

//Code in C++

#include <iostream>
using namespace std;
bool checkHeap(int arr[], int size)
{
    //traverse node till last internal node
    for (int i=0; i<=(size-2)/2; i++)
    {
        // If left child is greater, return false
        if (arr[2*i +1] > arr[i])
            return false;

        // If right child is greater, return false
        if (2*i+2 < size && arr[2*i+2] > arr[i])
            return false;
    }
    return true;
}

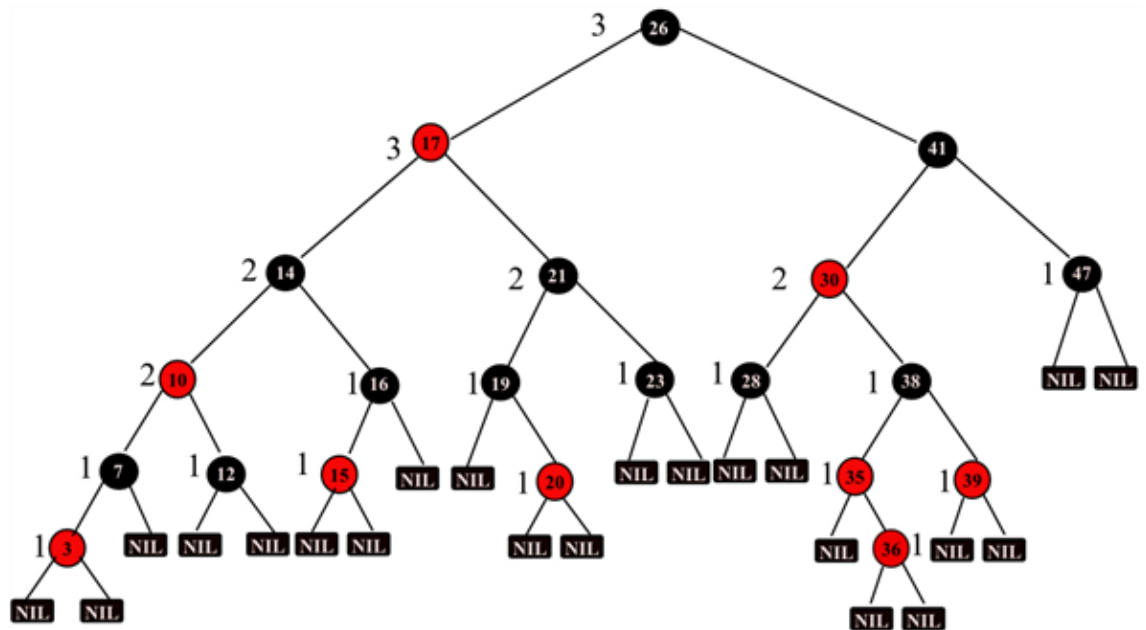
int main()
{
    int arr[]={16,14,10,8,7,9,3,2,4,1};
    int arr2[]={10,3,9,7,2,11,5,1,6};
    int n=10;
    bool res=checkHeap(arr,n);
    cout<<"Given array: ";
    for(int i=0;i<n;i++)
        cout<<arr[i]<<" ";

    if(res)
        cout<<"\nYES, heap"<<endl;
    else
        cout<<"\nNot a heap"<<endl;
    return 0;
}

C++ Tab Width: 8 Ln 42, Col 2 INS
```

```
Terminal
File Edit View Search Terminal Help
rnrorderhaug@ecc-f-03:~/Desktop/CS477/hw5$ ./5
Given arrays: 16 14 10 8 7 9 3 2 4 1
YES, heap
rnrorderhaug@ecc-f-03:~/Desktop/CS477/hw5$ ./5
Given arrays: 16 14 10 8 7 9 3 2 4 1
YES, heap
rnrorderhaug@ecc-f-03:~/Desktop/CS477/hw5$ g++ -o hw5.cpp
g++: fatal error: no input files
compilation terminated.
rnrorderhaug@ecc-f-03:~/Desktop/CS477/hw5$ ls
5 hw5.cpp
rnrorderhaug@ecc-f-03:~/Desktop/CS477/hw5$ g++ -o 5 hw5.cpp
rnrorderhaug@ecc-f-03:~/Desktop/CS477/hw5$ ./5
Given array: 16 14 10 8 7 9 3 2 4 1
Not a heap
rnrorderhaug@ecc-f-03:~/Desktop/CS477/hw5$ g++ -o 5 hw5.cpp
rnrorderhaug@ecc-f-03:~/Desktop/CS477/hw5$ ./5
Given array: 10 3 9 7 2 11 5 1 6 0
Not a heap
rnrorderhaug@ecc-f-03:~/Desktop/CS477/hw5$ g++ -o 5 hw5.cpp
rnrorderhaug@ecc-f-03:~/Desktop/CS477/hw5$ ./5
Given array: 10 3 9 7 2 11 5 1 6
Not a heap
rnrorderhaug@ecc-f-03:~/Desktop/CS477/hw5$
```

```
Terminal
File Edit View Search Terminal Help
Not a heap
rnrorderhaug@ecc-f-03:~/Desktop/CS477/hw5$ g++ -o 5 hw5.cpp
hw5.cpp: In function 'int main()':
hw5.cpp:36:12: error: 'i' was not declared in this scope
  cout<<arr2[i]<<" ";
             ^
rnrorderhaug@ecc-f-03:~/Desktop/CS477/hw5$ g++ -o 5 hw5.cpp
hw5.cpp: In function 'int main()':
hw5.cpp:36:12: error: 'i' was not declared in this scope
  cout<<arr2[i]<<" ";
             ^
rnrorderhaug@ecc-f-03:~/Desktop/CS477/hw5$ g++ -o 5 hw5.cpp
hw5.cpp: In function 'int main()':
hw5.cpp:36:12: error: 'i' was not declared in this scope
  cout<<arr2[i]<<" ";
             ^
rnrorderhaug@ecc-f-03:~/Desktop/CS477/hw5$ g++ -o 5 hw5.cpp
rnrorderhaug@ecc-f-03:~/Desktop/CS477/hw5$ ./5
Given arrays: 16 14 10 8 7 9 3 2 4 1
YES, heap
rnrorderhaug@ecc-f-03:~/Desktop/CS477/hw5$ ./5
Given arrays: 16 14 10 8 7 9 3 2 4 1
YES, heap
rnrorderhaug@ecc-f-03:~/Desktop/CS477/hw5$
```



4.

- a) After the key 36 is added to the tree, coloring it red will cause an error because rule 3 says if a node is red than both of its children must be black. If the node is colored black, then the black height property is violated as well, so in neither case is it a Red black Tree.
- b) Yes, it is still a RBT because it still satisfied all the properties of Red Black trees.

6.

6) © Find the smallest and the largest number of keys that a heap of height  $h$  can have

Smallest # of keys:  $\sum_{i=0}^h 2^i - 1 + 1$  Assuming height  $h$

$$= (2^h - 1) + 1$$

$$= 2^h - 1 + 1$$

$$= 2^h$$

Largest # of keys:  $\sum_{i=0}^{h+1} 2^i - 1$

$$= (2^{h+1} - 1)$$

Thus, the smallest and largest number of keys a heap of height  $h$  can have is  $2^h$  and  $2^{h+1} - 1$  respectively.

⑥ Prove that the height of a heap with  $n$  nodes is  $\lfloor \log_2 n \rfloor$

Assuming a heap of  $n$  nodes

$$2^h \leq n < 2^{h+1} - 1$$

$$\log 2^h \leq n < \log 2^{h+1} - 1$$

$$h \leq \log_2 n < h+1$$

Thus, the height of a heap with  $n$  nodes is  $\lfloor \log_2 n \rfloor$