CS163 – Data Structures & Algorithms

Lab 06
Red Black tree – Heap

Department of Software Engineering-FIT-VNU-HCMUS

1

Notes

Create a single solution/folder to store your source code in a week.

Then, create a project/sub-folder to store your source code of each assignment.

The source code in an assignment should have at least 3 files:

- A header file (.h): struct definition, function prototypes/definition.
- A source file (.cpp): function implementation.
- Another source file (.cpp): named YourID_Ex01.cpp, main function. Replace 01 by id of an assignment.

Make sure your source code was built correctly. Use many test cases to check your code before submitting to Moodle.

2

Content

In this lab, we will review the following topics:

- What is an AVL tree? How to manipulate it?
- What is a heap? How to manipulate it?

3

Assignments

A: YY: 01

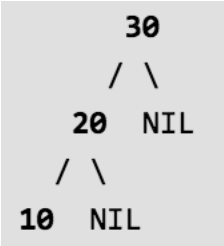
H: YY: 04

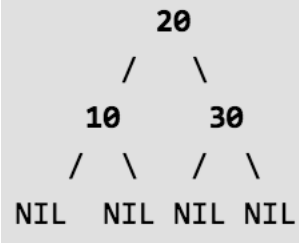
Visualization:

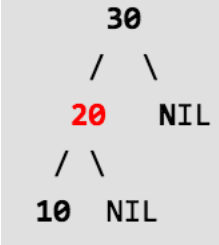
1. https://www.cs.usfca.edu/~galles/visualization/Algorithms.html
2. https://visualgo.net/en

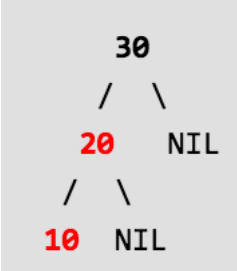
3.1. Assignment Red Black tree (Part 1)

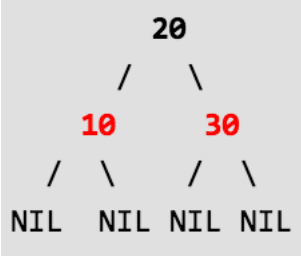
1. What is the properties of a Red Black tree? Give an example of Red Black tree.
2. Given examples of Red Black trees, each of them violates one of the Red Black tree properties.
3. Which trees are red black trees? If not, which property is violated?



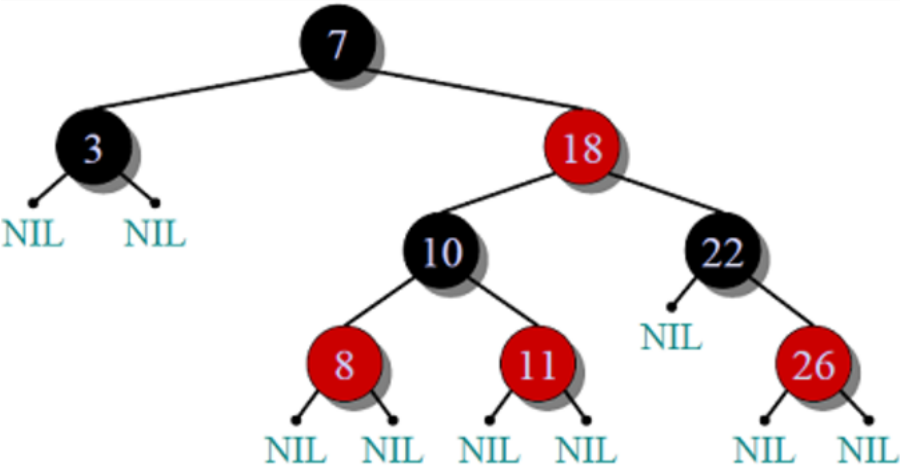








4. Calculate the black heights of all nodes in the below red back trees.

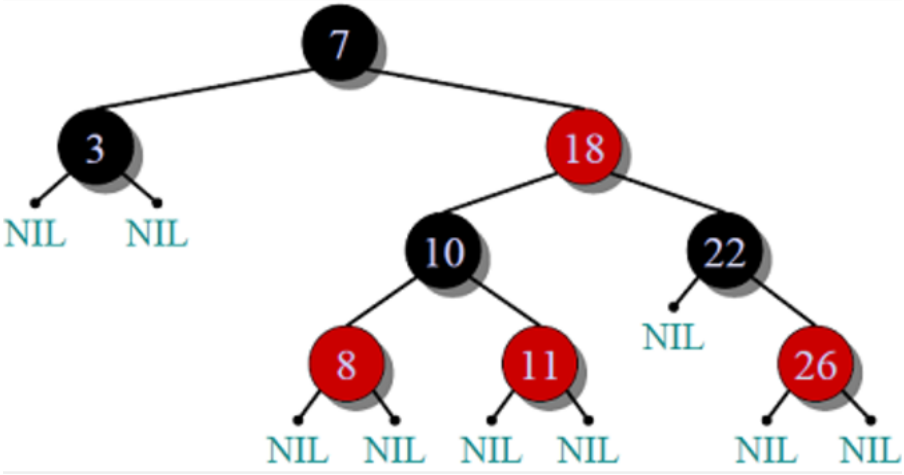


5. Is it possible to have all black nodes in a Red-Black tree? If yes, give an example. If no, give the reason.

3.2. Assignment Red Black tree (Part 2)

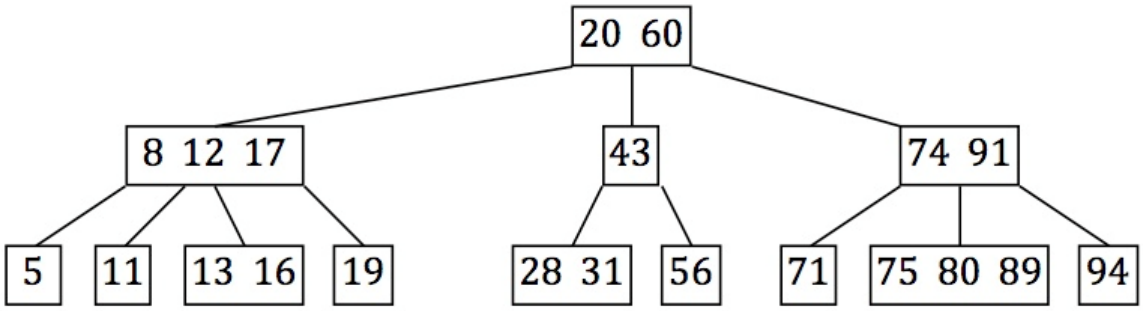
1. Merge red nodes into their black parents. Is it a 2-3-4 tree?

   Hint: https://stackoverflow.com/questions/35955246/converting-a-2-3-4-tree-into-a-red-black-tree

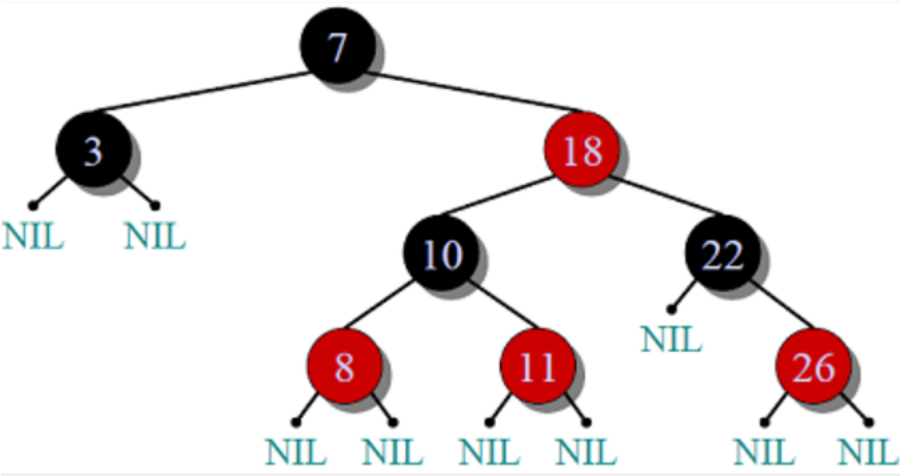


2. Convert the below 2-3-4 tree into a red black tree

Hint: https://stackoverflow.com/questions/35955246/converting-a-2-3-4-tree-into-a-red-black-tree



3. Insert 15 into the below red black tree.



3.3. Assignment Heap (Paper assignment)

1. What are the minimum and maximum numbers of elements in a heap of height h?
2. Where in a max-heap might the smallest element reside, assuming that all elements are distinct?
3. Is an array that is in sorted order a min-heap?
4. Is the array with values (23, 17, 14, 6, 13, 10, 1, 5, 7, 12) a max-heap?
5. Draw a heap (27,17,3,16,13,10,1,5,7,12,4,8,9,0). Then, illustrate the operation of MAX-HEAPIFY (A,2) on that heap. (index = 2, 0-based)
6. What is the effect of calling MAX-HEAPIFY(A,i) when the element A[i] is larger than its children? Explain your answer.
7. What is the effect of calling MAX-HEAPIFY(A,i) for i > A.heap-size/2? Explain your answer.
8. Illustrate the operation of BUILD-MAX-HEAP on the array A = (5,3,17,10,84,19,6,22,9).
9. Illustrate the operation of HEAPSORT on the array A = (5,13,2,25,7,17,20,8,4).
10. What is the running time of HEAPSORT on an array A of length n that is already sorted in increasing order? What about decreasing order?
11. Implement heapSort(a,n).

3.4. Assignment Heap (Coding assignment)

Implement the following functions in a min-heap or max-heap.

1. parent(a, n, i)
2. left(a, n, i)
3. right(a, n, i)
4. maxHeapify(a, n, i)
5. buildMaxHeap(a, n)
6. heapSort(a,n)