CSCI 2275 – Programming and Data structures

Instructor: Hoenigman

Recitation 7

Heaps and Testcases

Objectives:

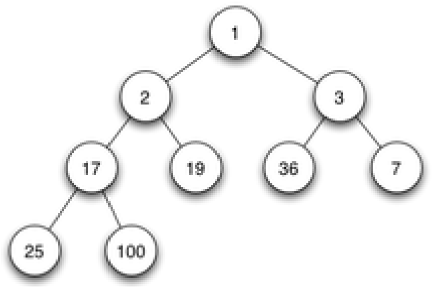
1. Heaps
2. Push and Pop Algorithm
3. Test cases
4. Exercise

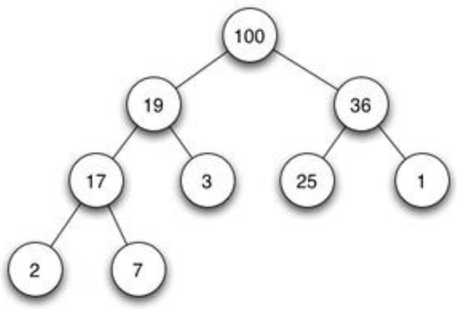
Heaps

A Binary Heap is a specialized ***tree-based*** data structure. It has the following properties:

● It is a ***complete tree***, meaning that all levels of the tree are completely filled except possibly the last level, where new nodes are inserted at the last level from length to right.

● It can be either a ***Min Heap*** or ***Max Heap***:





Implementation in Array:

Binary Heaps are usually implemented in an array. Why might this be?

The above min heap can be implemented in the following way:

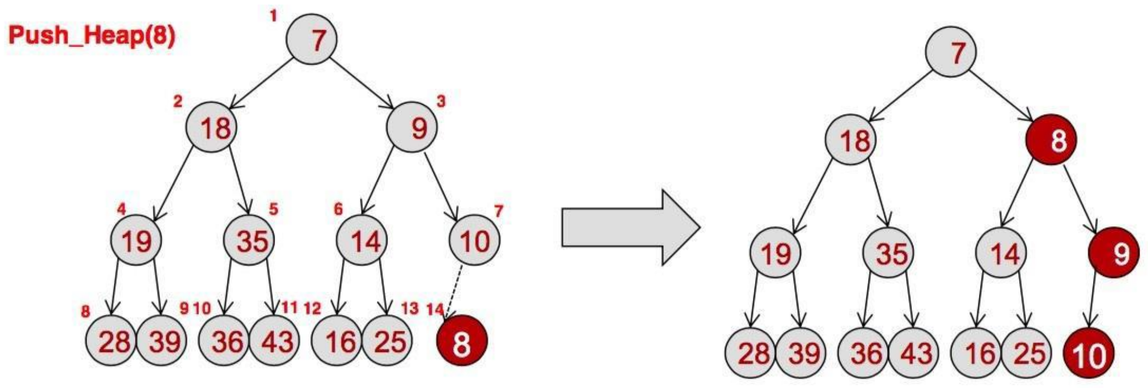
Node	1	2	3	17	19	36	7	25	100
Index	1	2	3	4	5	6	7	8	9

For a node at index ***x*** in the array:

1. Its parent is x/2
2. Its left child is 2*x
3. Its right child is 2*x+1

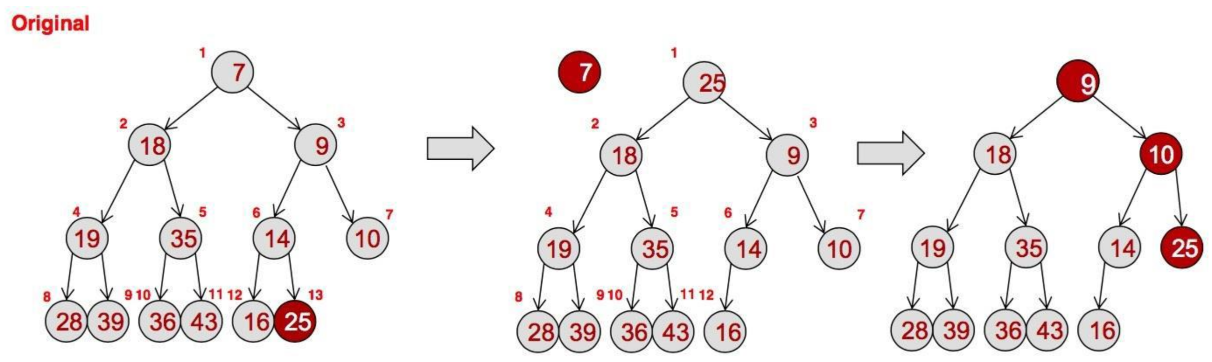
The Push Heap Algorithm:

The algorithm can be described as follows: insert the new node at the end of the array first and then reheap the newly inserted node (compare with its parent and find the right position).



The Pop Heap Algorithm:

The algorithm can be described as follows: remove the root node; move the node at the end of the array to the root; reheap the new root node(compare with its child and find the right position).



Test Cases

Write a function to return the type of a triangle based on the value of the length of 3 sides of a triangle. Let's make it a bit more easy, by assuming that test for input data type is already in place, so you receive only numeric values to work with.

The situation looks easy you go ahead and write the function which looks something like this –

Algorithm :

```
Input : 3 numeric values

Output : 1 string stating type of triangle

Function : triangleType(side1, side2, side3)

Start :

1. If side1 == side2 == side3

   Then Return "Equilateral Triangle"

2. Else if side1 == side2 or side1 == side3 or side2 == side3
```

```
  Then Return "Isosceles Triangle"
3. Else
  Return "Scalar Triangle"
Stop
```

Where will this code fail?

Exercise:

Write test cases for insertion in BST. Think about what can go wrong. Print the inorder and preorder traversal after every insert and check if that's what it should be.

(Test cases referenced from GeeksforGeeks)