# NPTEL Course: Programming, Data Structures and Algorithms in Python (*by* Prof. Madhvan Mukund)

*Tutorial (Week 6)*

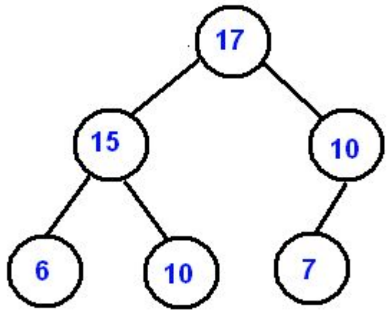
*Presented by*: Jivesh Dixit

## Problem 1 (a): Creating a binary max-Heap from the given sequence, adding and removing nodes and sorting using heap sort

*Approach:*

- ➢ *Heap is a special type of binary tree. In case of max-heap, parent must have larger value than its children on either side (left or right)*
- ➢ *For a node 'i', its parent will be node '(i-1)//2' and its children (left and right respectively) will be at nodes '2*i + 1' and '2*i + 2'.*
- ➢ *Left nodes must be completely filled before we can start filling right nodes.*
- ➢ *Following these set of rules and constraints we can create a binary max-heap.*
- ➢ *After deletion of root or insertion of node, we must maintain the structure of heap according to the constraints for max-heap.*
- ➢ *For sorting using heap-sort, in loop we can switch the root with last element, and then rearrange the heap around root till we get the sorted sequence.*

*Example: max-heap*

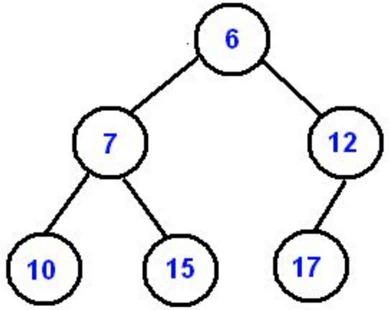


## Problem 1 (b): Creating a binary min-Heap from the given sequence, and sorting using heap sort.

*Approach:*

- ➢ *In case of min-heap, parent must have smaller value than its children on either side (left or right)*
- ➢ *Almost all other conditions remain the same as max-heap.*
- ➢ *Following the set of rules and constraints we can create a binary max-heap.*
- ➢ *For sorting using heap-sort, in loop we can switch the root with last element, and then rearrange the heap around root till we get the sorted sequence.*

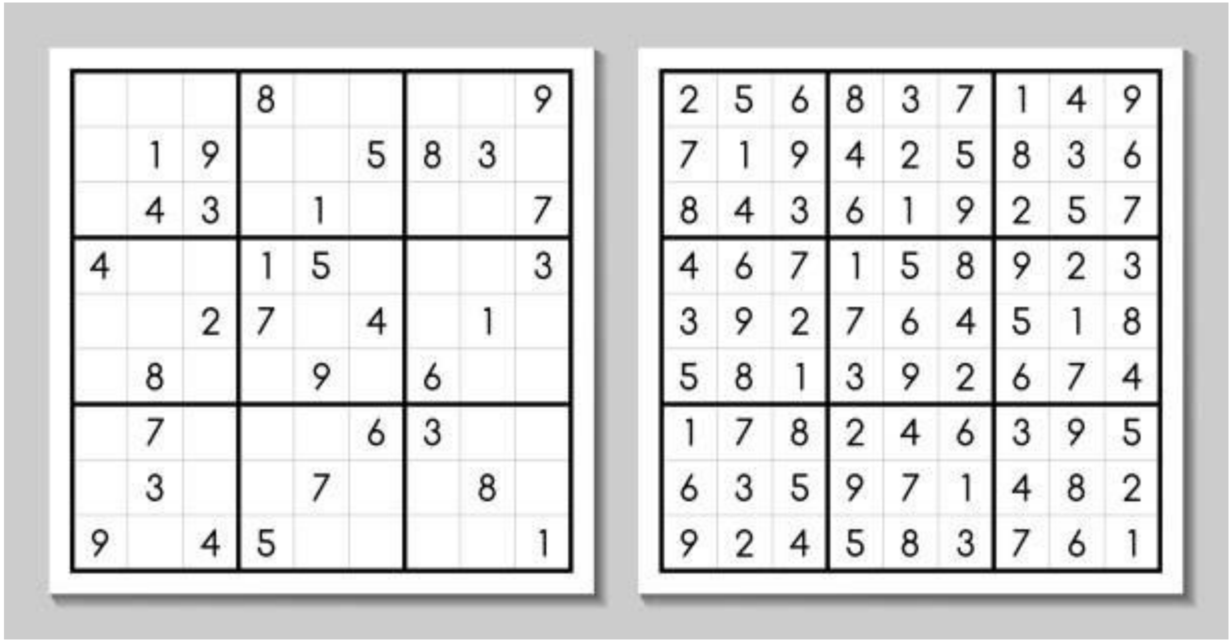
*Example: min-heap*



## Problem 2: Solving a given sudoku puzzle

Sudoku: *Sudoku is a puzzle in which player input numbers from ‘1’ to ‘9 into a grid consisting of nine squares subdivided into a further nine smaller squares in such a way that every number is inserted only once in each horizontal line, vertical line, and square.*

*Example:*
*https://ybshankar010.medium.com/solving-sudoku-using-backtracking-b2a4200daaac*



| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | 8 | | | | | 9 |
| | 1 | 9 | | | 5 | 8 | 3 | |
| | 4 | 3 | | 1 | | | | 7 |
| 4 | | | 1 | 5 | | | | 3 |
| | | 2 | 7 | | 4 | | 1 | |
| | 8 | | | 9 | | 6 | | |
| | 7 | | | | 6 | 3 | | |
| | 3 | | | 7 | | | 8 | |
| 9 | | 4 | 5 | | | | | 1 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 2 | 5 | 6 | 8 | 3 | 7 | 1 | 4 | 9 |
| 7 | 1 | 9 | 4 | 2 | 5 | 8 | 3 | 6 |
| 8 | 4 | 3 | 6 | 1 | 9 | 2 | 5 | 7 |
| 4 | 6 | 7 | 1 | 5 | 8 | 9 | 2 | 3 |
| 3 | 9 | 2 | 7 | 6 | 4 | 5 | 1 | 8 |
| 5 | 8 | 1 | 3 | 9 | 2 | 6 | 7 | 4 |
| 1 | 7 | 8 | 2 | 4 | 6 | 3 | 9 | 5 |
| 6 | 3 | 5 | 9 | 7 | 1 | 4 | 8 | 2 |
| 9 | 2 | 4 | 5 | 8 | 3 | 7 | 6 | 1 |

## Problem 2: Continued…

*Approach:*

- *As the numbers are to be filled in the puzzle are ‘1’ to ‘9’ inclusively. We can fill the unfilled entries of the puzzle of order N x N with ‘0’.*
- *Now we need to find the boxes with entry ‘0’ and check for the numbers ‘1’ to ‘9’ that can fit in without any disagreement across ‘row’ or ‘column’ or respective sub-square.*
- *Solving further, if we realize that we filled a box with some wrong entry, we can backtrack that box with zero and start over with remaining entries from ‘1’ to ‘9’.*