

CS 2302 Data Structures

Spring 2019

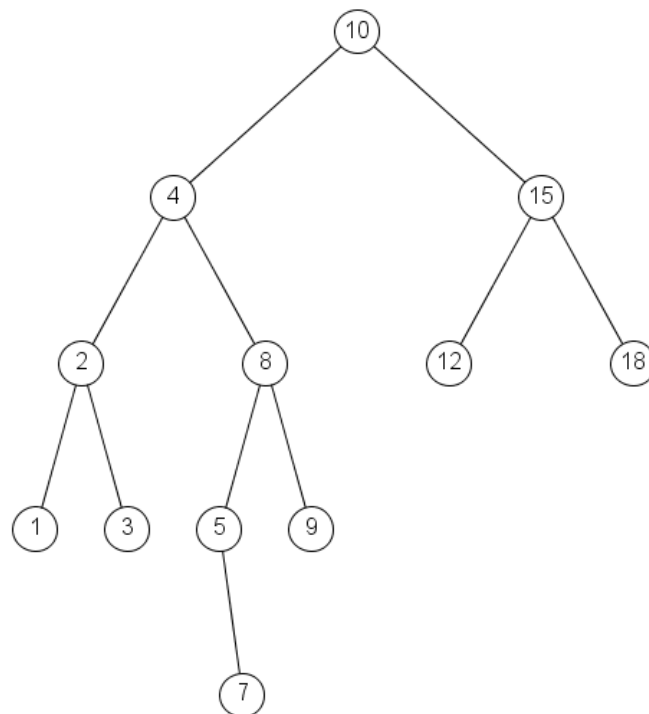
Lab 3 – Binary search trees

Due Friday, March 8

The program *bst.py* in the class webpage contains implementations of basic binary search tree operations, including insertion, deletion, search and display.

Modify the code to include the following operations:

- 1) Display the binary search tree as a figure, as shown below:



- 2) Iterative version of the search operation.
- 3) Building a balanced binary search tree given a sorted list as input. Note: this should not use the insert operation, the tree must be built directly from the list in $O(n)$ time.
- 4) Extracting the elements in a binary search tree into a sorted list. As above, this should be done in $O(n)$ time.
- 5) Printing the elements in a binary tree ordered by depth. The root has depth 0, the root's children have depth one, and so on. For example, for the tree in the figure, your program should output:

```
Keys at depth 0: 10
Keys at depth 1: 4 15
Keys at depth 2: 2 8 12 18
Keys at depth 3: 1 3 5 9
Keys at depth 4: 7
```

As usual, write a report describing your work.