

Lab 03

Binary search tree

Cảm ơn thầy Trần Duy Quang đã cung cấp template cho môn học



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.

Name of your submission: **StudentID_WEEK03_XX.zip**. XX: number of assignments you have done. XX: 00 – 99.

2

Content

In this lab, we will review the following topics:

- What is a binary search tree?
- How to insert, traverse and remove elements in a BST?

3

Assignments

A: XX: 01

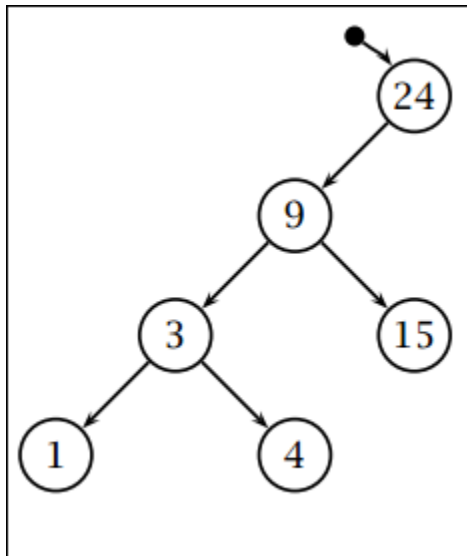
H: YY: 06

Visualization:

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

3.1. Paper assignment

1. Given the list of numbers as follow: 74, 12, 217, 36, 61, 77, 286, 153, 337, 93, 121, 47, 463, 248 and 146.
 - a. Build a BST for the above numbers.
 - b. Draw the BST after we remove node 146.
 - c. The above tree is not balanced. If you want to build a balanced BST tree, what if the orders of numbers in the above sequence? Draw a BST for new sequence of numbers.
2. Suppose we have integer values between 1 and 1000 in a BST and we want to search for 363. Which of the following can not be the sequence of keys examined. Why?
 - a. 2 252 401 398 330 363
 - b. 399 387 219 266 382 381 278 363
 - c. 3 923 220 911 244 898 258 362 363
 - d. 4 924 278 347 621 299 392 358 363
 - e. 5 925 202 910 245 363
3. Consider the following BST.



- a. List all possible insertion orders of the keys that could have produced this BST.
- b. Draw the same BST after the insertion of keys: 6, 45, 32, 98, 55, and 69, in this order.
- c. Draw the BST resulting from the deletion of keys 9 and 45 from the BST resulting from question b.
- d. Write an insertion order of the keys remaining in the BST after question 5c that would produce a balanced tree (i.e., a minimum-height tree).

3.2. Insertion

<https://www.hackerrank.com/challenges/binary-search-tree-insertion/problem>

3.3. Traversal

<https://www.hackerrank.com/challenges/tree-preorder-traversal/problem>

3.4. Top view traversal

<https://www.hackerrank.com/challenges/tree-top-view/problem>

3.5. Level order traversal

<https://www.hackerrank.com/challenges/tree-level-order-traversal/problem>

3.6. Univalued Binary Tree

<https://leetcode.com/problems/univalued-binary-tree/>