



HO CHI MINH UNIVERSITY OF SCIENCE
FACULTY OF INFORMATION TECHNOLOGY
SOFTWARE ENGINEERING DEPARTMENT
ADVANCED PROGRAM IN COMPUTER SCIENCE
COURSE: **DATA STRUCTURE**
LECTURER: Dr. ĐINH BÁ TIẾN

WEEK 03

BINARY SEARCH TREE (cont)

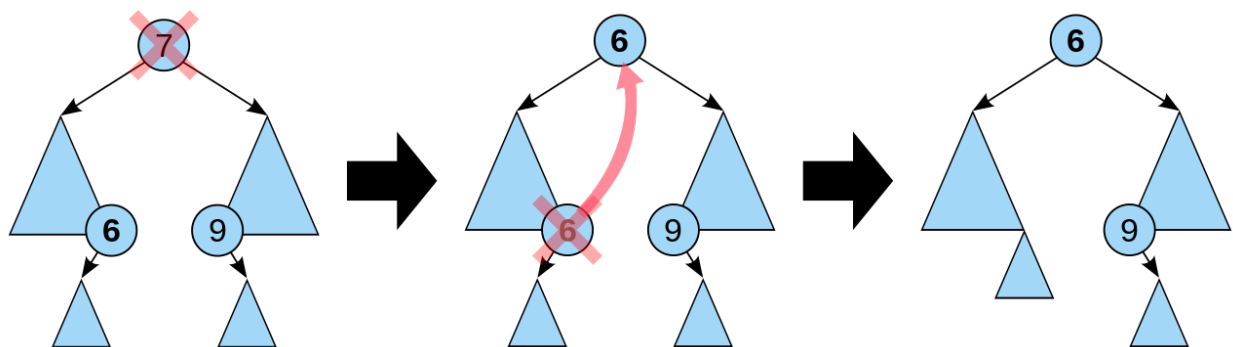
TRƯƠNG PHƯỚC LỘC
HỒ TUẤN THANH

HCMC, 2016

1 Binary search tree (cont)

1.1 Delete

- There are three possible cases to consider:
 - o Deleting a node with no children: simply remove the node from the tree.
 - o Deleting a node with one child: remove the node and replace it with its child.
 - o Deleting a node with two children: call the node to be deleted N. Do not delete N. Instead, choose either its in-order successor node or its in-order predecessor node, R. Copy the value of R to N, then recursively call delete on R until reaching one of the first two cases. If you choose in-order successor of a node, as right sub tree is not NIL (Our present case is node has 2 children), then its in-order successor is node with least value in its right sub tree, which will have at a maximum of 1 sub tree, so deleting it would fall in one of first 2 cases.



1.2 Reference

<https://www.cs.usfca.edu/~galles/visualization/BST.html>

2 Assignment

2.1 Assignment 01

- Write a small program to do the following tasks
 1. Load a list from “emotional-dictionary.txt” and create a binary search tree to store them.
 2. Search by key
 3. Search by content
 4. Remove an emotion (use both max of left and min of right)
 5. Manage favorite emotion
 - a. Add

- b. Remove
- c. Edit

2.2 Assignment 02

- Write a small program to do the following tasks
 1. Load a list from data of dictionary and create a binary search tree to store them.
 2. Search by key
 3. Search by content
 4. Remove a word
 5. Manage favorite emotion
 - a. Add
 - b. Remove
 - c. Edit
- Data (unicode): goldendict, colordict, *.dict, *.idx