System Software Experiment 1
Lecture 9

spring 2018

Jinkyu Jeong (jinkyu@skku.edu)
Computer Systems Laboratory
Sungyunkwan University
http://csl.skku.edu
Tree

• 데이터 저장에 용이
• 데이터 탐색에 걸리는 시간이 빠름 (than linked list)
Binary Tree

- Parent node
- Child node
- Sibling node
- Depth
- Level
- Height
Binary Tree using linked list

Shape of node

```
typedef struct _bTreeNode {
    int data;
    struct _bTreeNode *left;
    struct _bTreeNode *right;
} BTreeNode;
```
Binary Search Tree

- 노드의 왼쪽 subtree는 노드의 값보다 작은 값들을 갖는 노드들로 구성
- 노드의 오른쪽 subtree는 노드의 값보다 크거나 같은 값을 갖는 노드들로 구성

Today's exercise: make function for binary search tree
- Data insert
- Data delete
Insert Data into Binary Search Tree

<Insertion sequence : 10, 5, 15, 3, 8, 12, 2, 7, 9, 18, 19, 13>
Delete Data from Binary Search Tree

Case 1: if data is at termination node – just delete the node
<Delete node with data 19>
Delete Data from Binary Search Tree

Case 2: if data node has only one child – delete the node & link its parent node with child node

<Delete node with data 12>
Delete Data from Binary Search Tree

Case 2: if data node has two child – exchange smallest data with right subtree & delete minimum data node
<Delete node with data 5>