Lecture 32

Binary Trees Utilities

December 06, 2021
Monday

Recall the Terminologies

- Each node has to be reachable from the root through a unique sequences of edges called path.

- The level of a node is the length of the path from the root to the node plus 1, which is the number of nodes in the path.

- The height of a non-empty tree is the maximum level of a node in the tree.

Calculating the Height of a Node

```
int height ( Node p )
    {
        if( p == null )
            return 0;
        else
        {
            int left=height( p->left );
            int right=height(p->right);

            return Max (left, right) + 1;
        }
    }
```

Checking a BST?

```
int isBST(struct node* node)
{
  if (node == NULL)
    return 1;

  if (node->left != NULL && node->left->data > node->data)
    return 0;

  if (node->right != NULL && node->right->data < node->data)
    return 0;

  if (!isBST(node->left) || !isBST(node->right))
    return 0;


  return 1;
}
```

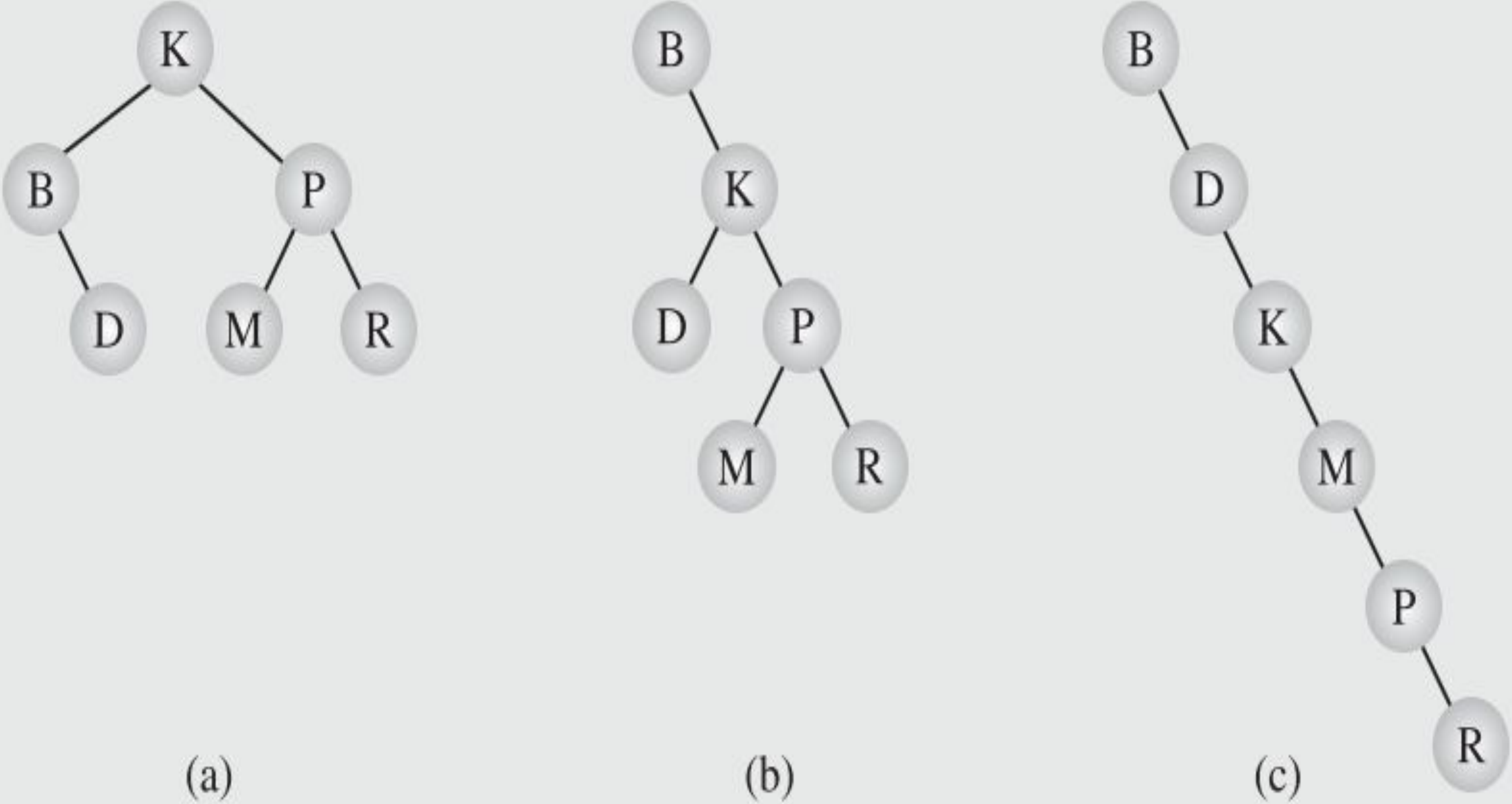
Left Heavy Tree Vs Right Heavy Tree

- **Left Heavy Tree**
  - If the left subtree of a given node has more height than the right subtree.
  - In terms of balancing factor, the factor should be negative.
  - Balancing factor = leftSubtreeHeight - rightSubtreeHeight.
  - Some literature also use node count to find if more nodes are present in the left subtree of a given node or right subtree.

The right Heavy Tree will be symmetric to this.

Balancing a Binary Tree

- Two arguments were presented in favor of trees
  a. They are well suited to represent the hierarchical structure
  b. The search process is much faster using trees instead of linked lists.
- The second argument, however, does not always hold.
- It all depends on what the tree looks like.


K
B
P
D
M
R
(a)
B
K
D
P
M
R
(b)
B
D
K
M
P
R
(c)

Balanced Tree

- A binary tree is height-balanced or simply balanced if the difference in height of both subtrees of any node in the tree is either zero or one.

- Also, a tree is considered perfectly balanced if it is balanced and all leaves are to be found on one level or two levels.

Why Balancing Matter ?

- For example, if 10,000 elements are stored in a perfectly balanced tree, then the tree is of height $\lceil \lg(10{,}001) \rceil = \lceil 13.289 \rceil = 14$.
- In practical terms, this means that if 10,000 elements are stored in a perfectly balanced tree, then at most 14 nodes have to be checked to locate a particular element.
- This is a substantial difference compared to the 10,000 tests needed in a linked list (in the worst case).

TREE BALANCING TECHNIQUES

- Some techniques require constantly restructuring the tree, when elements arrive and lead to unbalanced tree.

- Some of them consist of reordering the data themselves and then building a tree, if an ordering of the data guarantees that the resulting tree is balanced.

TECHNIQUE 1: REORDERING DATA FOR BALANCE

- When data arrive, store all of them in an array. After all the data arrive, sort the array using one of the efficient algorithms discussed already discussed.