Lecture 02/10/16

Lecturer: Xiaodi Wu

February 10th, 2016
Binary Search Tree

Recap

- We have defined Binary Search Tree (BST).
Recap

- We have defined Binary Search Tree (BST).
- Insertion and Removal operation on the BST. (Required in Lab 2)
Recap

- We have defined Binary Search Tree (BST).
- Insertion and Removal operation on the BST. (Required in Lab 2)
- Complexity of each operation is $O(h)$. 
Binary Search Trees

```
44
  /   \
17    88
/     /  \
32    65
/\    /\  \
28 29 54 82
  /\  /\  /\ \
29 76 80 78
   /\ /\ /\ /\ \
 78
```

Problems with BST

Complexity $O(h)$

$h$ could be $O(\log(n))$ or $O(n)$. The worst case complexity is $O(n)$. 

Candidate solution: AVL tree.
Problems with BST

Complexity $O(h)$

$h$ could be $O(\log(n))$ or $O(n)$. The worst case complexity is $O(n)$.

*Is there a way to force $h$ to be $O(\log(n))$ while still having efficient insertion and removal operation?*
Problems with BST

Complexity $O(h)$

$h$ could be $O(\log(n))$ or $O(n)$. The worst case complexity is $O(n)$.

*Is there a way to force $h$ to be $O(\log(n))$ while still having efficient insertion and removal operation?*

Candidate solution: **AVL tree**.