# Tutorial 6

## 1 Midterm

- Any questions about midterm?

## 2 BST Deletion

- Lazy deletion. Add a boolean "zombie" flag to each node Set to false initially on insertion, set to true on "deletion".
  - This works OK in short term, or if only few deletes, but performance degrades if tree is full of zombies
- A better way:
  1. Target node has no children.
     - Just delete the target node and send parent ptr to nullptr
  2. Target node has only one child.
     - Set the target node's parent to point to the only child. And delete thr target node.
  3. Target node has two children.
     - Find the replacement nodes
       - Biggest key down the left subtree(rightmost key in left subtree), or
       - Smallest key down the right subtree(leftmost key in right subtree)
     - Connect replacement parent's to its only child
     - Replace target node with replacement node
       - Set replacement node's left / right ptrs to values of target's left / right ptrs, respectively
       - Set target's parent to point to replacement node
     - Delete target node

## 3 BST Traversal

- Three kinds of traversal:
  1. Pre-Order
  2. In-Order
  3. Post-Order
- a trik to traversal a BST

# 4 Exercises

1. Write a function that prints out the contents of a BST using Pre-Order Traversal. For simplicity assume the keys are ints.

2. Write a function that prints out the contents of a BST using Post-Order Traversal. For simplicity assume the keys are ints.

3. Write a function that determines whether a Binary Tree(assume duplicated values are allowed) is a BST or not, returning true if it is and false otherwise.