Binary Search Trees Notes – Handout
Computer Science 223p
Python Programming

The code listings in this handout are taken from the identified pages of Introduction to Algorithms, 3rd Ed. by Cormen et al. This is an excellent reference book and all computer science students are encouraged to acquire their own copy.

```
// p.286
Inorder-Tree-Walk(x)
  Inorder-Tree-Walk(x.left)
  print x.key
  Inorder-Tree-Walk(x.right)


// p.290
Tree-Search(x, k)
  if x == NIL or k == x.key
    return x
  if k < x.key
    return Tree-Search(x.left, k)
  else
    return Tree-Search(x.right, k)


// p. 291
Iterative-Tree-Search(x, k)
  while x ≠ NIL and k ≠ x.key
    if k < x.key
      x = x.left
    else
      x = x.right
  return x


// p. 291
Tree-Minimum(x)
  while x.left ≠ NIL
    x = x.left
  return x


// p. 291
Tree-Maximum(x)
  while x.right ≠ NIL
    x = x.right
  return x


// p. 292
Tree-Successor(x)
  if x.right ≠ NIL
    return Tree-Minimum(x.right)
  y = x.p
  while y ≠ NIL and x == y.right
    x = y
    y = y.p
```

```
    return y

// p. 294
Tree-Insert(T, z)
  y = NIL
  x = T.root
  while x ≠ NIL
    y = x
    if z.key < x.key
      x = x.left
    else
      x = x.right
  z.p = y
  if y == NIL
    T.root = z
  elseif y.key > z.key
    y.left = z
  else
    y.right = z

// p.296
Transplant(T, u, v)
  if u.p == NIL
    T.root = v
  elseif u == u.p.left
    u.p.left = v
  else
    u.p.right = v
  if v ≠ NIL
    v.p = u.p

// p.298
Tree-Delete(T, z)
  if z.left == NIL
    Transplant(T, z, z.right)
  elseif z.right == NIL
    Transplant(T, z, z.left)
  else
    y = Tree-Minimum(z.right)
    if y.p ≠ z
      Transplant(T, y, y.right)
      y.right = z.right
      y.right.p = y
    Transplant(T, z, y)
    y.left = z.left
    y.left.p = y
```