Dictionaries

Represents a (mutable) set S of elements with keys

Operations

- `insert(D, x)`
  - add an element `x`
- `remove(D, x)`
  - remove `x` from the set
- `search(D, x)`
  - check if an element with key `x` is in the set, and return it if found

Binary Search Trees

BST Property

For each node, *keys in left subtree <= root key <= keys in right subtree*.

Traversals

- in-order:
  1. traverse left subtree in-order
  2. visit root
  3. traverse right subtree in-order
- pre-order:
  1. visit root
  2. traverse left subtree pre-order
  3. traverse right subtree pre-order
- post-order:

- traverse left subtree post-order
- traverse right subtree post-order
- visit root

Rotations

...

Set Operations on BSTs

`search(D, x)`

```
def search(D, x):
    if (x == D.root):
        return D.root
    elif (x < D.root):
        return search(D.left_child, x)
    else:
        return search(D.right_child, x)
```