

## Module 9 Homework: Binary Search Trees

Add some functionality to a binary search tree (BST) mapping.

Starter code for a BST is provided for you in the file `BSTMap.py`. **All of your coding for this assignment should be in the file `MyBSTMap.py`**, but feel free to reference the code in `BSTMap.py` (it has two classes you inherit - you will probably find it instructive to know how they work).

### Equality

Implement a method to compare two BSTs. Two BSTs are equal if both contain exactly the same k-v pairs and have the same shape.

This method will be called by the user using a tree object (and so should be defined in the tree class), but you will likely find it necessary to define equality for nodes and their recursive subtrees (e.g. within the node class) as well.

### Building Trees

We can build trees from lists that give pre-ordered or post-ordered traversal of that tree. Consider the following tree of `k:v` pairs, where the values are all `str(k)`:

```
#      3:'3'
#      /
#     1:'1'
#    /  \
# 0:'0' 2:'2'
```

Traversals of this tree yields the following lists of `k:v` tuples:

- pre-order: [(3,'3'), (1,'1'), (0,'0'), (2,'2')]
- post-order: [(0,'0'), (2,'2'), (1,'1'), (3,'3')].

Implement two functions:

- `frompreorder` - generates a BST from a list of pre-ordered `k, v` tuples
  - only takes the list as input
  - returns the generated BST
- `frompostorder` - generates a BST from a list of post-ordered `k, v` tuples
  - only takes the list as input
  - returns the generated BST

### Submitting

At a minimum, submit the following files:

- `BSTMap.py`
  - No need to modify, but it is imported in your code
- `MyBSTMap.py`
  - Your work will all be done here

Students must submit to Mimir individually by the due date (typically, the first Wednesday after this module opens at 11:59 pm) to receive credit.

### Grading

- 30 - Equal
- 35 - `frompreorder`
- 35 - `frompostorder`

## Feedback

If you have any feedback on this assignment, please leave it [here](#).

We check this feedback regularly, and it has resulted in many improvements.