Testing

We strongly encourage testing your code yourself. The given tests for this project are not comprehensive tests: it is possible to write incorrect code that passes them all (but not get full score).

Things that you might consider testing for include: anything that we specify in the comments or in this document that a method should do that you don't see a test already testing for, and any edge cases that you can think of. Think of what valid inputs might break your code and cause it not to perform as intended, and add a test to make sure things are working.

We've put together a video to introduce you to the testing framework.

To help you get started, here is one case that is *not* in the given tests (and will be included in the hidden tests): everything should work properly even if we delete everything from the BPlusTree. For example, after everything is deleted and a new iterator is created, the new iterator should immediately return false for `hasNext`.

(Note that we don't allow mutating the tree during a scan, so the behavior of an iterator created before everything was deleted is undefined, and can be handled however you like or not at all).

To add a unit test, open up the appropriate test file (in `src/test/java/edu/berkeley/cs186/database/index`) and simply add a new method to the file with a `@Test` annotation, for example:

```
@Test
public void testEverythingDeleted() {
    // your test code here
}
```

Many test classes have some setup code done for you already: take a look at other tests in the file for an idea of how to write the test code.

Previous
Your Tasks

Next
Submitting the Assignment

Last updated 5 months ago