Exercise 11.06: remove() and Coverage

Look at the code coverage for the remove() method on the projector. The diamond markers (in order) have the following messages:

1st diamond: 1 of 2 branches missed
2nd diamond: 1 of 2 branches missed
3rd diamond: 3 of 4 branches missed
4th diamond: 1 of 2 branches missed

The test method is:

```
@Test
public void testRemoveAtIdx() {
    LinkedIntList list = new LinkedIntList();
    list.add(23);
    assertEquals(1, list.size());
    try {
        list.remove(1);
        fail();
    } catch (IndexOutOfBoundsException e) {
        assertEquals(1, list.size());
    }
}
```

Answer the following questions about the above scenario:

Your username (**sesmith5@ncsu.edu**) will be recorded when you submit this form. Not **sesmith5**? Sign out
* Required

What are the EclEmma notifications telling you about the code coverage of the remove() method? *

What can be done to obtain full condition coverage on the method (what modifications need to be made to the test code so that all conditional predicates are executed on both their true and false paths)? *

Does the explanation and syntax highlighting given by EclEmma make the problem clear? Are there, in your opinion, any better ways of conveying this type of information? *

Unity id(s) of person(s) you worked with.

☑ Send me a copy of my responses.

Submit

Never submit passwords through Google Forms.

Powered by Google Docs

Report Abuse - Terms of Service - Additional Terms