Testing

Albert Ye

August 29, 2022

1 Ad Hoc testing

Most naive way to test is to just check manually and output something when it is wrong. Like I'd do in competitive programming.

Use `org.junit.Assert.assertArrayEquals` for a simpler way to check if two arrays are the same, and if so, to check where the differences are at.

This method has to be run with expected first and input second.

We can find one string instead of an entire array to get the current smallest.

1.1 Assertions

1. assertEquals
2. assertArrayEquals
3. assertFalse/assertTrue
4. assertNotNull
5. a lot of other things

2 JUnit

Give an annotation `org.JUnit.Test` before a function and then delete the `static` and then delete the main function because JUnit doesn't run as an executable java class.

This is why we need IntelliJ I guess, it's too difficult to set all of this up.

2.1 Even Better JUnit

Need

```
import org.junit.Test; // don't need to run @org.junit.Test just @Test
import static org.junit.Assert; // just need to run Assert.assertEquals() etc
```

and then we won't even need the ugly org.junit.* headers behind everything related to JUnit.

Unlike in Python, all Java libraries are in the system without imports, you just import to make calling classes in the libraries less verbosely (even though everything Java is verbose)