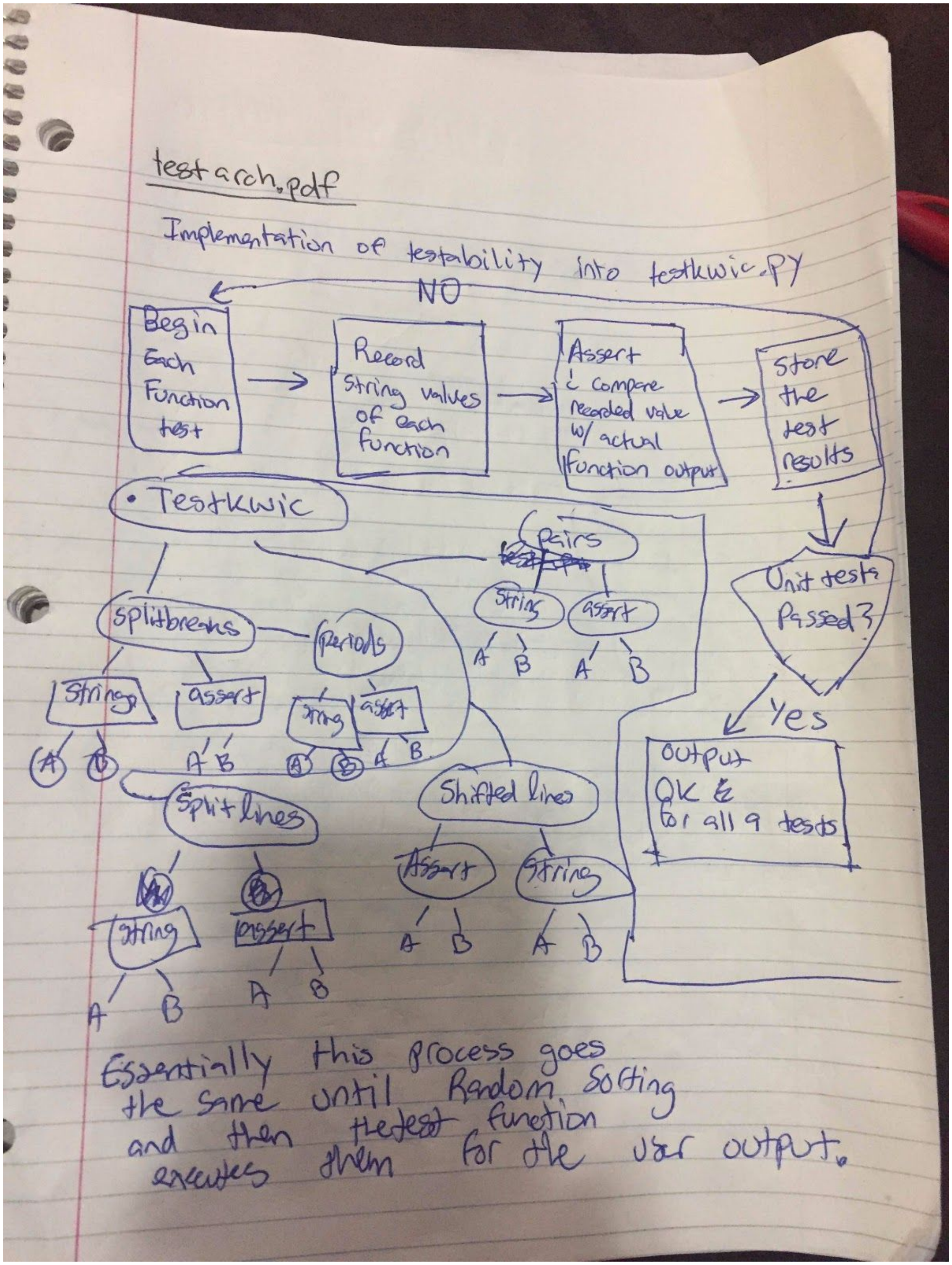
testarch.pdf



testarch.pdf

Implementation of testability into testkwic.py

Essentially this process goes the same until Random Sorting and then thetest function executes them for the user output.

Unittest Testability

```
Import unittest
Class TestKwic (unittest.TestCase):
    def test_splitbreaks(self):
        String = "test string. \nwill it split? \n Prob not."
        String B = "this test String, Has no endlines"
       self.assertEqual(splitBreaks(string, False), {'    ']}
       self.assertEqual(splitBreaks(string B, False), {'   ']})

   def test_splitbreaks_periods(self):
      String = "test String, Lets see if it splits?"

      String B = "Will a string with no endlines split"
      self.assertEqual (splitBreaks(string, True) ['   ']})
      self.assertEqual(splitBreaks (stringB, True)['   ']})

    def thetest ():
        testMain = Unittest.TestSuite ()
        testMain.addTest (TestKwic('test_splitbreaks'))
        testMain.addTest (TestKwic('test_splitbreaks_periods'))
        testMain.addTest (TestKwic('split_lines))
           return testMain

   def test_splitLines(Self)
     lines~~string~~ = ['Will the test!', 'String even', 'split?']
     lines B = ['Hello String with no endlines,']
     self.assertEqual(splitLines(lines), [['   ', ]]]
     self.assertEqual (splitLines(linesB), [{'', '', ]})
```

keep adding for each new test function