

Report Lab5

李子强 11510352

Exercise1

Test Case	Duration	Result
PhiladelphiaTest (Lab5)	20ms	Failed
testIsItSunny	4ms	Passed
testFailsPassword	15ms	Failed
testIsPassword	0ms	Passed
testFailsItSunny	1ms	Failed

```
D:\Java\jdk1.8.0_131\bin\java ...  
java.lang.AssertionError:  
Expected :654321  
Actual   :123456  
<Click to see difference>  
  
<1 internal call>  
at org.junit.Assert.failNotEquals(Assert.java:834) <2 internal calls>  
at Lab5.PhiladelphiaTest.testFailsPassword(PhiladelphiaTest.java:25) <22 internal calls>  
  
java.lang.AssertionError <4 internal calls>  
at Lab5.PhiladelphiaTest.testFailsItSunny(PhiladelphiaTest.java:14) <22 internal calls>  
  
Process finished with exit code -1
```

- Add a test in which an assertion as the picture shows.
- Write a new test that throws an exception

Exercise2

Test Case	Duration	Result
ArrayListTest (Lab5)	9ms	Passed
testAdd	8ms	Passed
testGet	0ms	Passed
testClear	0ms	Passed
testNotContains	0ms	Passed
testRemoveObject	0ms	Passed
testIndexOf	1ms	Passed

```
D:\Java\jdk1.8.0_131\bin\java ...  
Process finished with exit code 0
```

- 添加并测试 `clear`, `contains`, `get` 方法。

Exercise3

Test Case	Duration	Result
TimeParserTest (Lab5)	14ms	Passed
testPath	13ms	Passed
testnoon	0ms	Passed
testzero	0ms	Passed
testmidnight	0ms	Passed
testBranch	1ms	Passed

```
D:\Java\jdk1.8.0_131\bin\java ...  
Process finished with exit code 0
```

- 测试 `parseTimeToSeconds` 方法。

Exercise4

Test Case	Duration	Result
MinHeapArrayInvariant1Test (Lab5)	40ms	Passed
testPop	19ms	Passed
testClear	1ms	Passed
testWithRandomRemoves	11ms	Passed
testWithRandomAdds	9ms	Passed

```
D:\Java\jdk1.8.0_131\bin\java ...  
Process finished with exit code 0
```

- 测试堆的边界条件: `array[n] <= array[2n+1]` and `array[n] <= array[2n+2]`

```
private boolean invariantHolds() {
    Integer top = heap.peek();
    if (top == null) {
        return true;
    }
    Integer[] contents = new Integer[heap.size()];
    contents = heap.toArray(contents);
    for (int i = 0; i < (heap.size() - 1) / 2; i++) {
        if (contents[i] > contents[2 * i + 1] || contents[i] > contents[2 * i + 2]) {
            System.out.println("Whoops!");
            return false;
        }
    }
    return true;
}
```