./test/prime.py

Killed 16 out of 23 mutants

Timeouts

Mutants that made the test suite take a lot longer so the tests were killed.

Mutant 32

```
--- ./test/prime.py
+++ ./test/prime.py
@@ -11,7 +11,7 @@
             for p in d[q]:
                 d.setdefault(p + q, []).append(p)
             del d[q]
-        q += 1
+        q += 2


 import math
```

Survived

Survived mutation testing. These mutants show holes in your test suite.

Mutant 24

```
--- ./test/prime.py
+++ ./test/prime.py
@@ -2,7 +2,7 @@
 # Code by David Eppstein, UC Irvine, 28 Feb 2002
 # http://code.activestate.com/recipes/117119/
 def gen_prime_eratosthenes():
-    d, q = {}, 2
+    d, q = None
     while True:
         if q not in d:
             yield q
```

Mutant 25

```
--- ./test/prime.py
+++ ./test/prime.py
@@ -3,7 +3,7 @@
 # http://code.activestate.com/recipes/117119/
 def gen_prime_eratosthenes():
     d, q = {}, 2
-    while True:
+    while False:
         if q not in d:
             yield q
             d[q * q] = [q]
```

Mutant 26

```
--- ./test/prime.py
+++ ./test/prime.py
@@ -4,7 +4,7 @@
 def gen_prime_eratosthenes():
     d, q = {}, 2
     while True:
-        if q not in d:
+        if q  in d:
             yield q
             d[q * q] = [q]
         else:
```

Mutant 28

```
--- ./test/prime.py
+++ ./test/prime.py
@@ -6,7 +6,7 @@
     while True:
         if q not in d:
             yield q
-            d[q * q] = [q]
+            d[q * q] = None
         else:
             for p in d[q]:
                 d.setdefault(p + q, []).append(p)
```

Mutant 35

```
--- ./test/prime.py
+++ ./test/prime.py
@@ -16,7 +16,7 @@

 import math
 def is_prime_simple(n):
-    q = int(math.sqrt(n)+1)
+    q = None
     for x in range(2, q):
         if n % x == 0:
             return False
```

Mutant 44

```
--- ./test/prime.py
+++ ./test/prime.py
@@ -24,5 +24,5 @@

 import re
 def is_prime_re(n):
-    return re.compile(r'^1?$|^(11+)\1+$').match('1' * n) is  None
+    return re.compile(r'^1?$|^(11+)\1+$').match('1' / n) is  None
```