

# Unit Testing Progress Report for EasyGraph

July 1, 2025

## 1 Coverage Statistics

The table below compares the coverage before and after the recent testing improvements:

Table 1: Test Coverage Improvement

Metric	Before	After
Total Number of Tests	261	786
Classes Module Coverage	57%	67%
Functions Module Coverage	38%	67%

## 2 Visualizations

### 2.1 Growth in Total Number of Tests

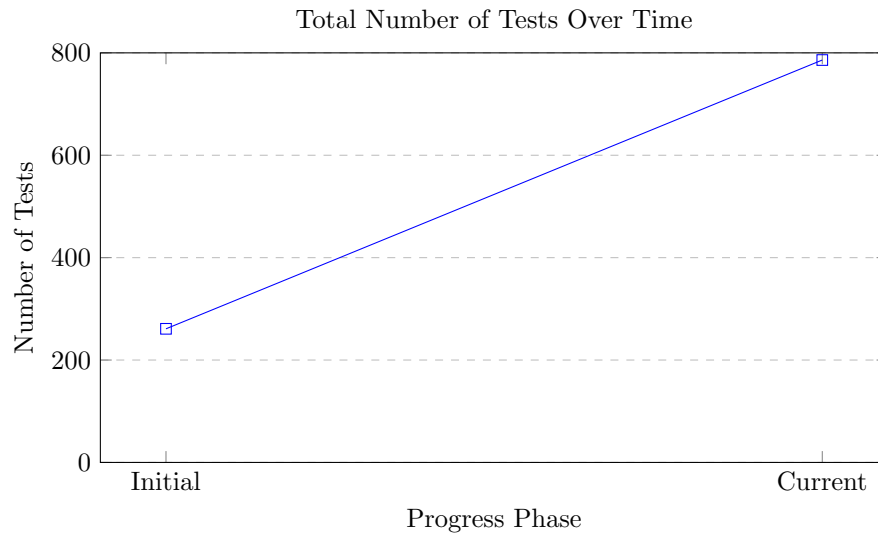


Figure 1: Growth in Total Unit Tests

## 2.2 Coverage Improvement by Module

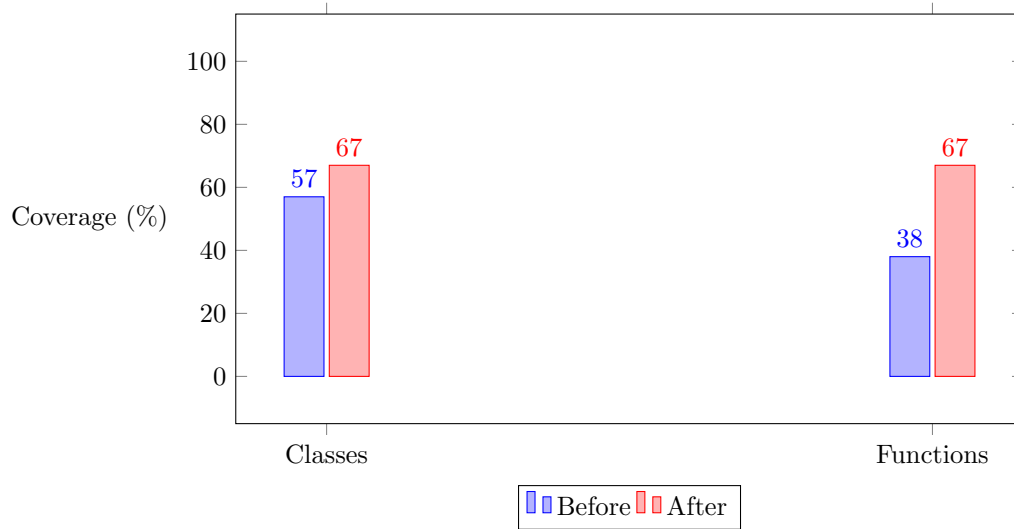


Figure 2: Code Coverage Improvement by Module

## 3 Coverage Report Screenshots

### 3.1 Total tests comparison

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
tests/test_convert.py .... [100%]

===== short test summary info =====
SKIPPED [1] readwrite/tests/test_graphviz.py:7: could not import 'pygraphviz': No module named 'pygraphviz'
SKIPPED [1] classes/tests/test_hypergraph.py:973: skip
SKIPPED [1] readwrite/tests/test_graphml.py:1104: could not import 'lxml.etree': No module named 'lxml'
SKIPPED [1] readwrite/tests/test_graphml.py:1110: could not import 'lxml.etree': No module named 'lxml'
SKIPPED [1] readwrite/tests/test_graphml.py:1122: could not import 'lxml.etree': No module named 'lxml'
SKIPPED [1] readwrite/tests/test_graphml.py:1128: could not import 'lxml.etree': No module named 'lxml'
SKIPPED [1] readwrite/tests/test_graphml.py:1141: could not import 'lxml.etree': No module named 'lxml'
SKIPPED [1] readwrite/tests/test_graphml.py:1186: could not import 'lxml.etree': No module named 'lxml'
SKIPPED [1] readwrite/tests/test_graphml.py:1212: could not import 'lxml.etree': No module named 'lxml'
SKIPPED [1] readwrite/tests/test_graphml.py:1229: could not import 'lxml.etree': No module named 'lxml'
SKIPPED [1] readwrite/tests/test_graphml.py:1244: could not import 'lxml.etree': No module named 'lxml'
SKIPPED [1] readwrite/tests/test_graphml.py:1257: could not import 'lxml.etree': No module named 'lxml'
SKIPPED [1] readwrite/tests/test_graphml.py:1270: could not import 'lxml.etree': No module named 'lxml'
SKIPPED [1] readwrite/tests/test_graphml.py:1284: could not import 'lxml.etree': No module named 'lxml'
SKIPPED [1] readwrite/tests/test_graphml.py:1295: could not import 'lxml.etree': No module named 'lxml'
SKIPPED [1] readwrite/tests/test_graphml.py:1319: could not import 'lxml.etree': No module named 'lxml'
SKIPPED [1] readwrite/tests/test_graphml.py:1363: could not import 'lxml.etree': No module named 'lxml'
SKIPPED [1] readwrite/tests/test_graphml.py:1422: could not import 'lxml.etree': No module named 'lxml'
SKIPPED [1] readwrite/tests/test_graphml.py:1438: could not import 'lxml.etree': No module named 'lxml'
SKIPPED [1] readwrite/tests/test_graphml.py:1454: could not import 'lxml.etree': No module named 'lxml'
SKIPPED [1] readwrite/tests/test_graphml.py:1464: could not import 'lxml.etree': No module named 'lxml'
SKIPPED [1] readwrite/tests/test_graphml.py:1477: could not import 'lxml.etree': No module named 'lxml'
===== 261 passed, 22 skipped, 62 warnings in 7.26s =====

(venv) same@node0:~/easyGraphTests/Easy-Graph/easygraph$
```

Figure 3: Previous total tests

```
===== 786 passed, 22 skipped, 88 warnings in 33.74s =====
** On entry to DGEEV parameter number 5 had an illegal value
** On entry to DGEEV parameter number 5 had an illegal value
(venv) same@node0:~/easyGraphTests/Easy-Graph/easygraph$
```

Figure 4: Current total tests

### 3.2 Classes module test coverage comparison

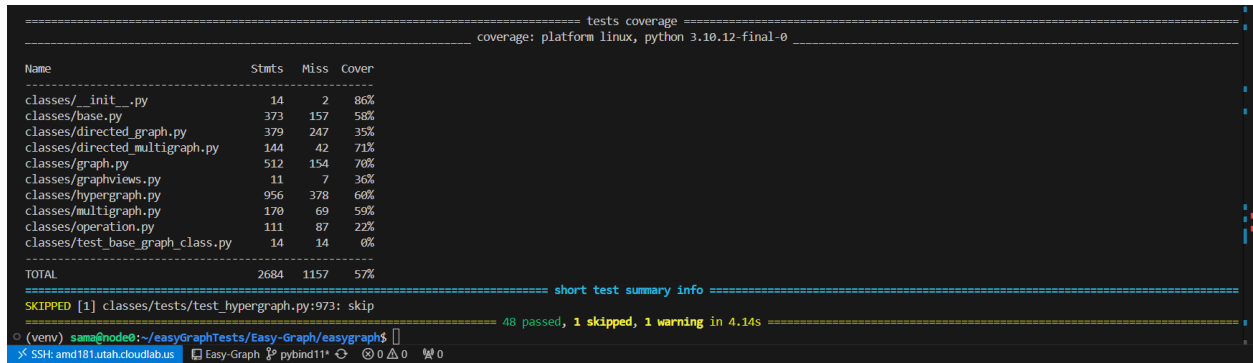
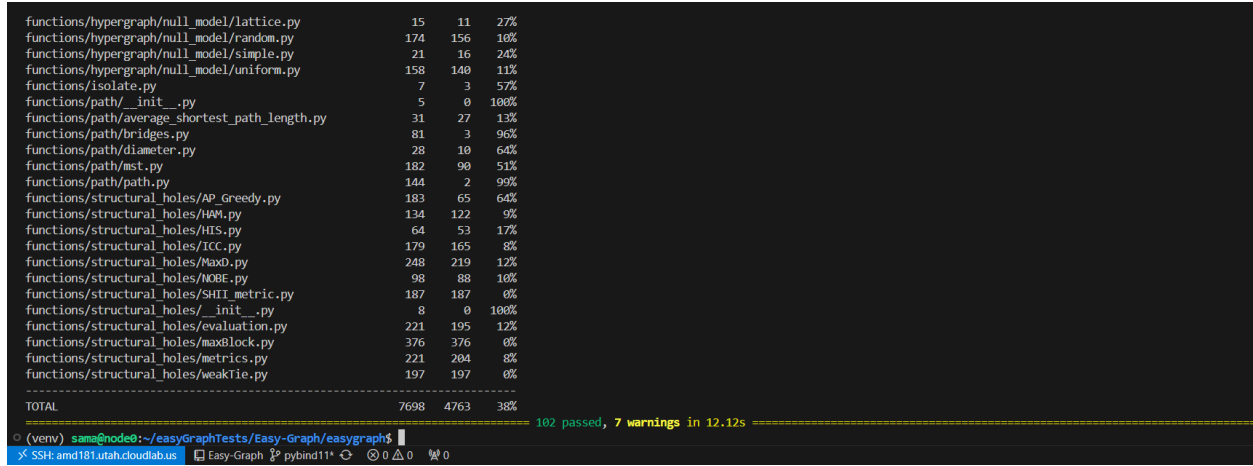


Figure 5: Previous classes test coverage



Figure 6: Current classes test coverage

### 3.3 Functions module test coverage comparisson

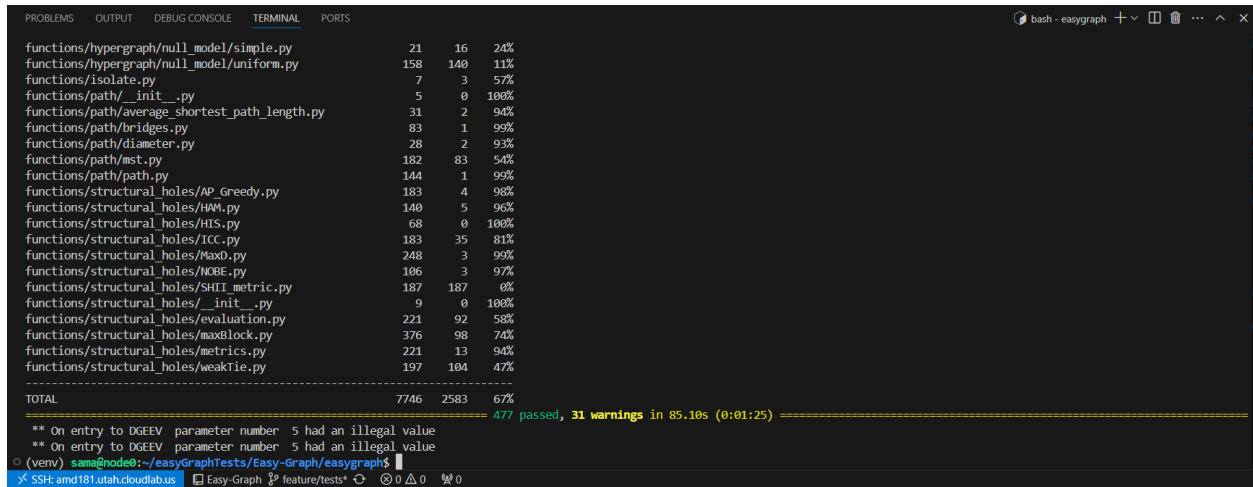


functions/hypergraph/null_model/lattice.py	15	11	27%
functions/hypergraph/null_model/random.py	174	156	10%
functions/hypergraph/null_model/simple.py	21	16	24%
functions/hypergraph/null_model/uniform.py	158	140	11%
functions/isolate.py	7	3	57%
functions/path/_init_.py	5	0	100%
functions/path/average_shortest_path_length.py	31	27	13%
functions/path/bridges.py	81	3	96%
functions/path/diameter.py	28	10	64%
functions/path/mst.py	182	90	51%
functions/path/path.py	144	2	99%
functions/structural_holes/AP_Greedy.py	183	65	64%
functions/structural_holes/HAM.py	134	122	9%
functions/structural_holes/HIS.py	64	53	17%
functions/structural_holes/ICC.py	179	165	8%
functions/structural_holes/MaxD.py	248	219	12%
functions/structural_holes/NOBE.py	98	88	10%
functions/structural_holes/SHII_metric.py	187	187	0%
functions/structural_holes/_init_.py	8	0	100%
functions/structural_holes/evaluation.py	221	195	12%
functions/structural_holes/maxBlock.py	376	376	0%
functions/structural_holes/metrics.py	221	204	8%
functions/structural_holes/weakTie.py	197	197	0%
TOTAL	7698	4763	38%

===== 102 passed, 7 warnings in 12.12s =====

○ (venv) same@node0:~/easyGraphTests/Easy-Graph/easygraphs\$

Figure 7: Previous functions test coverage



functions/hypergraph/null_model/simple.py	21	16	24%
functions/hypergraph/null_model/uniform.py	158	140	11%
functions/isolate.py	7	3	57%
functions/path/_init_.py	5	0	100%
functions/path/average_shortest_path_length.py	31	2	94%
functions/path/bridges.py	83	1	99%
functions/path/diameter.py	28	2	93%
functions/path/mst.py	182	83	54%
functions/path/path.py	144	1	99%
functions/structural_holes/AP_Greedy.py	183	4	98%
functions/structural_holes/HAM.py	140	5	96%
functions/structural_holes/HIS.py	68	0	100%
functions/structural_holes/ICC.py	183	35	81%
functions/structural_holes/MaxD.py	248	3	99%
functions/structural_holes/NOBE.py	106	3	97%
functions/structural_holes/SHII_metric.py	187	187	0%
functions/structural_holes/_init_.py	9	0	100%
functions/structural_holes/evaluation.py	221	92	58%
functions/structural_holes/maxBlock.py	376	98	74%
functions/structural_holes/metrics.py	221	13	94%
functions/structural_holes/weakTie.py	197	104	47%
TOTAL	7746	2583	67%

===== 477 passed, 31 warnings in 85.10s (0:01:25) =====

\*\* On entry to DGEEV parameter number 5 had an illegal value  
\*\* On entry to DGEEV parameter number 5 had an illegal value

○ (venv) same@node0:~/easyGraphTests/Easy-Graph/easygraphs\$

Figure 8: Current functions test coverage

## 4 Conclusion

The number of unit tests in EasyGraph increased by over  $3\times$ , from 261 to 786. Test coverage for the **classes** module rose from 57% to 67%, and for the **functions** module from 38% to 67%. These improvements significantly enhance code reliability, encourage safe refactoring, and contribute to the long-term quality of the project.