Sample output from my solution to Problem #1:
(yours should match the format: the times depend on your machine’s speed).

```
Spanning Tree of size 1000
Analysis of 5 timings
avg = 0.07839   min = 0.07630  max = 0.08122  span = 6.3%

   Time Ranges
7.63e-02<>7.68e-02[ 40.0%]|**************************************************
7.68e-02<>7.73e-02[  0.0%]|
7.73e-02<>7.78e-02[ 20.0%]|*************************
7.78e-02<>7.83e-02[  0.0%]|
7.83e-02<>7.88e-02[  0.0%]|A
7.88e-02<>7.92e-02[  0.0%]|
7.92e-02<>7.97e-02[  0.0%]|
7.97e-02<>8.02e-02[  0.0%]|
8.02e-02<>8.07e-02[ 20.0%]|*************************
8.07e-02<>8.12e-02[  0.0%]|
8.12e-02<>8.17e-02[ 20.0%]|*************************

Spanning Tree of size 2000
Analysis of 5 timings
avg = 0.16485   min = 0.15975  max = 0.16932  span = 5.8%

   Time Ranges
1.60e-01<>1.61e-01[ 20.0%]|**************************************************
1.61e-01<>1.62e-01[  0.0%]|
1.62e-01<>1.63e-01[ 20.0%]|**************************************************
1.63e-01<>1.64e-01[  0.0%]|
1.64e-01<>1.65e-01[  0.0%]|
1.65e-01<>1.65e-01[ 20.0%]|**************************************************A
1.65e-01<>1.66e-01[  0.0%]|
1.66e-01<>1.67e-01[  0.0%]|
1.67e-01<>1.68e-01[ 20.0%]|**************************************************
1.68e-01<>1.69e-01[  0.0%]|
1.69e-01<>1.70e-01[ 20.0%]|**************************************************

Spanning Tree of size 4000
Analysis of 5 timings
avg = 0.34483   min = 0.33914  max = 0.35152  span = 3.6%

   Time Ranges
3.39e-01<>3.40e-01[ 40.0%]|**************************************************
3.40e-01<>3.42e-01[  0.0%]|
3.42e-01<>3.43e-01[  0.0%]|
3.43e-01<>3.44e-01[  0.0%]|
3.44e-01<>3.45e-01[ 20.0%]|*************************A
3.45e-01<>3.47e-01[  0.0%]|
3.47e-01<>3.48e-01[  0.0%]|
3.48e-01<>3.49e-01[  0.0%]|
3.49e-01<>3.50e-01[ 20.0%]|*************************
3.50e-01<>3.52e-01[  0.0%]|
3.52e-01<>3.53e-01[ 20.0%]|*************************

Spanning Tree of size 8000
Analysis of 5 timings
avg = 0.76256   min = 0.72598  max = 0.86182  span = 17.8%

   Time Ranges
7.26e-01<>7.40e-01[ 40.0%]|**************************************************
```

```
7.40e-01<>7.53e-01[ 40.0%]|**************************************************
7.53e-01<>7.67e-01[  0.0%]|A
7.67e-01<>7.80e-01[  0.0%]|
7.80e-01<>7.94e-01[  0.0%]|
7.94e-01<>8.07e-01[  0.0%]|
8.07e-01<>8.21e-01[  0.0%]|
8.21e-01<>8.35e-01[  0.0%]|
8.35e-01<>8.48e-01[  0.0%]|
8.48e-01<>8.62e-01[  0.0%]|
8.62e-01<>8.75e-01[ 20.0%]|*************************

Spanning Tree of size 16000
Analysis of 5 timings
avg = 1.59328   min = 1.55889  max = 1.63900  span = 5.0%

   Time Ranges
1.56e+00<>1.57e+00[ 20.0%]|*************************
1.57e+00<>1.57e+00[ 40.0%]|**************************************************
1.57e+00<>1.58e+00[  0.0%]|
1.58e+00<>1.59e+00[  0.0%]|
1.59e+00<>1.60e+00[  0.0%]|A
1.60e+00<>1.61e+00[  0.0%]|
1.61e+00<>1.61e+00[  0.0%]|
1.61e+00<>1.62e+00[  0.0%]|
1.62e+00<>1.63e+00[  0.0%]|
1.63e+00<>1.64e+00[ 20.0%]|*************************
1.64e+00<>1.65e+00[ 20.0%]|*************************

Spanning Tree of size 32000
Analysis of 5 timings
avg = 3.48097   min = 3.41479  max = 3.52660  span = 3.2%

   Time Ranges
3.41e+00<>3.43e+00[ 20.0%]|**************************************************
3.43e+00<>3.44e+00[  0.0%]|
3.44e+00<>3.45e+00[  0.0%]|
3.45e+00<>3.46e+00[  0.0%]|
3.46e+00<>3.47e+00[  0.0%]|
3.47e+00<>3.48e+00[ 20.0%]|**************************************************A
3.48e+00<>3.49e+00[ 20.0%]|**************************************************
3.49e+00<>3.50e+00[ 20.0%]|**************************************************
3.50e+00<>3.52e+00[  0.0%]|
3.52e+00<>3.53e+00[  0.0%]|
3.53e+00<>3.54e+00[ 20.0%]|**************************************************

Spanning Tree of size 64000
Analysis of 5 timings
avg = 7.60396   min = 7.46765  max = 7.71107  span = 3.2%

   Time Ranges
7.47e+00<>7.49e+00[ 20.0%]|**************************************************
7.49e+00<>7.52e+00[  0.0%]|
7.52e+00<>7.54e+00[  0.0%]|
7.54e+00<>7.57e+00[  0.0%]|
7.57e+00<>7.59e+00[ 20.0%]|**************************************************
7.59e+00<>7.61e+00[ 20.0%]|**************************************************A
7.61e+00<>7.64e+00[  0.0%]|
7.64e+00<>7.66e+00[  0.0%]|
7.66e+00<>7.69e+00[ 20.0%]|**************************************************
7.69e+00<>7.71e+00[  0.0%]|
7.71e+00<>7.74e+00[ 20.0%]|**************************************************
```

```
Spanning Tree of size 128000
Analysis of 5 timings
avg = 16.69047   min = 16.11995  max = 17.67365  span = 9.3%

   Time Ranges
1.61e+01<>1.63e+01[ 20.0%]|*************************
1.63e+01<>1.64e+01[ 20.0%]|*************************
1.64e+01<>1.66e+01[  0.0%]|
1.66e+01<>1.67e+01[ 40.0%]|**************************************************A
1.67e+01<>1.69e+01[  0.0%]|
1.69e+01<>1.71e+01[  0.0%]|
1.71e+01<>1.72e+01[  0.0%]|
1.72e+01<>1.74e+01[  0.0%]|
1.74e+01<>1.75e+01[  0.0%]|
1.75e+01<>1.77e+01[  0.0%]|
1.77e+01<>1.78e+01[ 20.0%]|*************************
```

Sample output from my solution to Problem #2:
(yours should match the format: the times/counts depend on your machine’s speed and the random graph created).

```
Sat Dec  1 08:15:25 2018    profile15K

         2323438 function calls (2308437 primitive calls) in 1.703 seconds

   Ordered by: call count
   List reduced from 18 to 10 due to restriction <10>

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
   599625    0.031    0.000    0.031    0.000 {built-in method builtins.len}
   314812    0.266    0.000    0.490    0.000 graph.py:23(__getitem__)
   309702    0.202    0.000    0.202    0.000 equivalence.py:28(_compress_to_root)
   299813    0.143    0.000    0.874    0.000 graph_goody.py:26(<genexpr>)
   299813    0.168    0.000    0.715    0.000 graph.py:125(__iter__)
   299812    0.208    0.000    0.224    0.000 graph.py:12(legal_tuple)
   139852    0.063    0.000    0.248    0.000 equivalence.py:60(in_same_class)
  15002/1    0.489    0.000    1.306    1.306 {built-in method builtins.sorted}
    15000    0.005    0.000    0.005    0.000 equivalence.py:19(add_singleton)
    14999    0.016    0.000    0.033    0.000 equivalence.py:68(merge_classes_containing)
```

```
Sat Dec  1 08:15:25 2018    profile15K

         2323438 function calls (2308437 primitive calls) in 1.703 seconds

   Ordered by: internal time
   List reduced from 18 to 10 due to restriction <10>

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
  15002/1    0.489    0.000    1.306    1.306 {built-in method builtins.sorted}
   314812    0.266    0.000    0.490    0.000 graph.py:23(__getitem__)
   299812    0.208    0.000    0.224    0.000 graph.py:12(legal_tuple)
   309702    0.202    0.000    0.202    0.000 equivalence.py:28(_compress_to_root)
   299813    0.168    0.000    0.715    0.000 graph.py:125(__iter__)
   299813    0.143    0.000    0.874    0.000 graph_goody.py:26(<genexpr>)
        1    0.073    0.073    1.671    1.671 graph_goody.py:24(spanning_tree)
   139852    0.063    0.000    0.248    0.000 equivalence.py:60(in_same_class)
        1    0.032    0.032    1.703    1.703 <string>:1(<module>)
   599625    0.031    0.000    0.031    0.000 {built-in method builtins.len}
```