

Sample output from my solution to Problem #1: (yours should match the format: the times depend on your machine's speed). Some machines are so fast that small problem sizes cannot be analyzed correctly.

```
find_influencers of size 100
Analysis of 5 timings
0 time-elapsed; cannot show analysis
```

```
find_influencers of size 200
Analysis of 5 timings
0 time-elapsed; cannot show analysis
```

```
find_influencers of size 400
Analysis of 5 timings
avg = 0.01250   min = 0.00000   max = 0.01562   span = 125.0%
```

```
Time Ranges
0.00e+00<>1.56e-03[ 20.0%] | *****
1.56e-03<>3.13e-03[  0.0%] |
3.13e-03<>4.69e-03[  0.0%] |
4.69e-03<>6.25e-03[  0.0%] |
6.25e-03<>7.81e-03[  0.0%] |
7.81e-03<>9.37e-03[  0.0%] |
9.38e-03<>1.09e-02[  0.0%] |
1.09e-02<>1.25e-02[  0.0%] |
1.25e-02<>1.41e-02[  0.0%] | A
1.41e-02<>1.56e-02[  0.0%] |
1.56e-02<>1.72e-02[ 80.0%] | *****
```

```
find_influencers of size 800
Analysis of 5 timings
avg = 0.03750   min = 0.03125   max = 0.04688   span = 41.7%
```

```
Time Ranges
3.12e-02<>3.28e-02[ 60.0%] | *****
3.28e-02<>3.44e-02[  0.0%] |
3.44e-02<>3.59e-02[  0.0%] |
3.59e-02<>3.75e-02[  0.0%] |
3.75e-02<>3.91e-02[  0.0%] | A
3.91e-02<>4.06e-02[  0.0%] |
4.06e-02<>4.22e-02[  0.0%] |
4.22e-02<>4.38e-02[  0.0%] |
4.37e-02<>4.53e-02[  0.0%] |
4.53e-02<>4.69e-02[  0.0%] |
4.69e-02<>4.84e-02[ 40.0%] | *****
```

```
find_influencers of size 1600
Analysis of 5 timings
avg = 0.14688   min = 0.14062   max = 0.15625   span = 10.6%
```

```
Time Ranges
1.41e-01<>1.42e-01[ 60.0%] | *****
1.42e-01<>1.44e-01[  0.0%] |
1.44e-01<>1.45e-01[  0.0%] |
1.45e-01<>1.47e-01[  0.0%] |
1.47e-01<>1.48e-01[  0.0%] | A
1.48e-01<>1.50e-01[  0.0%] |
1.50e-01<>1.52e-01[  0.0%] |
1.52e-01<>1.53e-01[  0.0%] |
1.53e-01<>1.55e-01[  0.0%] |
```

```

1.55e-01<>1.56e-01[ 0.0%]|
1.56e-01<>1.58e-01[ 40.0%]| *****

```

find_influencers of size 3200

Analysis of 5 timings

avg = 0.57812 min = 0.56250 max = 0.59375 span = 5.4%

Time Ranges

```

5.62e-01<>5.66e-01[ 20.0%]| *****
5.66e-01<>5.69e-01[ 0.0%]|
5.69e-01<>5.72e-01[ 0.0%]|
5.72e-01<>5.75e-01[ 0.0%]|
5.75e-01<>5.78e-01[ 0.0%]|
5.78e-01<>5.81e-01[ 60.0%]| *****A
5.81e-01<>5.84e-01[ 0.0%]|
5.84e-01<>5.88e-01[ 0.0%]|
5.88e-01<>5.91e-01[ 0.0%]|
5.91e-01<>5.94e-01[ 0.0%]|
5.94e-01<>5.97e-01[ 20.0%]| *****

```

find_influencers of size 6400

Analysis of 5 timings

avg = 2.33125 min = 2.23438 max = 2.40625 span = 7.4%

Time Ranges

```

2.23e+00<>2.25e+00[ 20.0%]| *****
2.25e+00<>2.27e+00[ 0.0%]|
2.27e+00<>2.29e+00[ 0.0%]|
2.29e+00<>2.30e+00[ 0.0%]|
2.30e+00<>2.32e+00[ 20.0%]| *****
2.32e+00<>2.34e+00[ 20.0%]| *****A
2.34e+00<>2.35e+00[ 0.0%]|
2.35e+00<>2.37e+00[ 0.0%]|
2.37e+00<>2.39e+00[ 20.0%]| *****
2.39e+00<>2.41e+00[ 0.0%]|
2.41e+00<>2.42e+00[ 20.0%]| *****

```

find_influencers of size 12800

Analysis of 5 timings

avg = 9.44375 min = 9.35938 max = 9.54688 span = 2.0%

Time Ranges

```

9.36e+00<>9.38e+00[ 40.0%]| *****
9.38e+00<>9.40e+00[ 0.0%]|
9.40e+00<>9.42e+00[ 20.0%]| *****
9.42e+00<>9.43e+00[ 0.0%]|
9.43e+00<>9.45e+00[ 0.0%]| A
9.45e+00<>9.47e+00[ 0.0%]|
9.47e+00<>9.49e+00[ 0.0%]|
9.49e+00<>9.51e+00[ 0.0%]|
9.51e+00<>9.53e+00[ 0.0%]|
9.53e+00<>9.55e+00[ 20.0%]| *****
9.55e+00<>9.57e+00[ 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 Mar 13 19:36:41 2021 test_profile

2783477 function calls (2783476 primitive calls) in 1.015 seconds

Ordered by: call count

List reduced from 108 to 20 due to restriction <20>

ncalls	totttime	percall	cumtime	percall	filename:lineno(function)
817404	0.096	0.000	0.096	0.000	influence.py:69 (<lambda>)
408702	0.227	0.000	0.323	0.000	adjustablepriorityqueue.py:22 (_trichotomy)
341099	0.024	0.000	0.024	0.000	{built-in method builtins.len}
278023	0.034	0.000	0.034	0.000	adjustablepriorityqueue.py:44 (_parent)
272668	0.064	0.000	0.084	0.000	adjustablepriorityqueue.py:60 (_in_heap)
219454	0.103	0.000	0.103	0.000	adjustablepriorityqueue.py:64 (_swap)
151553	0.021	0.000	0.021	0.000	adjustablepriorityqueue.py:29 (_left_child)
121115	0.015	0.000	0.015	0.000	adjustablepriorityqueue.py:36 (_right_child)
42438	0.044	0.000	0.385	0.000	adjustablepriorityqueue.py:134 (updated)
37353	0.204	0.000	0.571	0.000	adjustablepriorityqueue.py:95 (_percolate_down)
25083	0.100	0.000	0.273	0.000	adjustablepriorityqueue.py:72 (_percolate_up)
18424	0.005	0.000	0.011	0.000	adjustablepriorityqueue.py:170 (is_empty)
18424	0.004	0.000	0.005	0.000	adjustablepriorityqueue.py:177 (size)
9999	0.012	0.000	0.509	0.000	adjustablepriorityqueue.py:121 (remove)
9999	0.002	0.000	0.002	0.000	{built-in method math.ceil}
9999	0.001	0.000	0.001	0.000	{method 'pop' of 'list' objects}
1575	0.000	0.000	0.000	0.000	{method 'add' of 'set' objects}
12	0.000	0.000	0.000	0.000	{method 'rstrip' of 'str' objects}
7	0.000	0.000	0.000	0.000	{method 'join' of 'str' objects}
6	0.000	0.000	0.000	0.000	<frozen importlib._bootstrap>:222(_verbose_message)

Sat Mar 13 19:36:41 2021 test_profile

2783477 function calls (2783476 primitive calls) in 1.015 seconds

Ordered by: internal time

List reduced from 108 to 20 due to restriction <20>

ncalls	totttime	percall	cumtime	percall	filename:lineno(function)
408702	0.227	0.000	0.323	0.000	adjustablepriorityqueue.py:22 (_trichotomy)
37353	0.204	0.000	0.571	0.000	adjustablepriorityqueue.py:95 (_percolate_down)
219454	0.103	0.000	0.103	0.000	adjustablepriorityqueue.py:64 (_swap)
25083	0.100	0.000	0.273	0.000	adjustablepriorityqueue.py:72 (_percolate_up)
817404	0.096	0.000	0.096	0.000	influence.py:69 (<lambda>)
272668	0.064	0.000	0.084	0.000	adjustablepriorityqueue.py:60 (_in_heap)
42438	0.044	0.000	0.385	0.000	adjustablepriorityqueue.py:134 (updated)
1	0.043	0.043	1.015	1.015	influence.py:65 (find_influencers3)
278023	0.034	0.000	0.034	0.000	adjustablepriorityqueue.py:44 (_parent)
341099	0.024	0.000	0.024	0.000	{built-in method builtins.len}
151553	0.021	0.000	0.021	0.000	adjustablepriorityqueue.py:29 (_left_child)
121115	0.015	0.000	0.015	0.000	adjustablepriorityqueue.py:36 (_right_child)
9999	0.012	0.000	0.509	0.000	adjustablepriorityqueue.py:121 (remove)
1	0.010	0.010	0.014	0.014	influence.py:68 (<dictcomp>)
18424	0.005	0.000	0.011	0.000	adjustablepriorityqueue.py:170 (is_empty)
18424	0.004	0.000	0.005	0.000	adjustablepriorityqueue.py:177 (size)
9999	0.002	0.000	0.002	0.000	{built-in method math.ceil}
1	0.002	0.002	0.055	0.055	adjustablepriorityqueue.py:52 (_heapify)
1	0.001	0.001	0.002	0.002	influence.py:67 (<setcomp>)
1	0.001	0.001	0.001	0.001	adjustablepriorityqueue.py:16 (<dictcomp>)