Problem 1b. Enter the Time and compute the Ratio of Times to two decimal places (x.xx)

Graph Size	Time for Computing find_influencers	Ratio of Time: Size 2N/Size N
100	Cannot analysis	No ratio for first graph size
200	0.00313	No ratio
400	0.01250	3.99
800	0.04688	3.75
1,600	0.15937	3.4
3,200	0.66563	4.18
6,400	2.74063	4.12
12,800	11.0406	4.03

Approximate the complexity class for the find_influencers function based on the data above.

Answer: O(N**2)

Using the last measurement in the table above, predict how long it would take to find the influencers in a graph with 1,000,000 nodes. Write the first number less than 10 followed by the appropriate unit: seconds, minutes, hours, days, weeks, months (assume 30 day months), years, etc. E.g., 1,000 seconds would be written as .277 hours. Show your work.

T(15625)~15.56875s

T(31250)~62.275s

T(62500)~249.1s

T(125000)~996.4s

T(250000)~3985.6s

T(500000)~15942.4s

T(1000000)~63769.6s=0.74 days

Problem 2b. Answer each of the following question based on the profiles produced when running `find_influencers3` on a 10,000 node /50,00 edge random graph sorted by `ncalls` for parts 2 and 3; sorted by `tottime` for parts 1 and 4.

1) What function/method/lambda takes the most `tottime` to execute?

Answer: **`_trichotomy`**

2) What built-in function/method/lambda is called the most times?

Answer: `math.ceil`

3) What function/method/lambda defined in `influence.py` is called the most times?

Answer: `lambda x : infl_dict[x]`

4) How many (the fewest number) functions/methods/lambdas consume 50% of the execution time? Name them.

Answer: 3; **`_trichotomy, _percolate_down, _percolate_up`**