Problem 1b. Enter data in the **Time to Compute** closest_2d for each **List Size** and compute the **Ratio of Times** for each size and its previous (half as big) size, to 5 decimal places (x.xxxxx).

List Size	**Time to Compute** closest_2d	**Ratio of Time: Size 2N/Size N**
100	0.00313	No previous N to Compute Ratio
200	0.00937	2.99361
400	0.02813	3.00213
800	0.06563	3
1,600	0.13750	2.33309
3,200	0.31875	2.31818
6,400	0.69375	2.17647
12,800	1.58125	2.27927
25,600	3.49688	2.21146

Approximate the complexity class for the closest_2d function based on the data above.

Answer: O(N)

Using the last measurement in the table above, predict how long it would take to find the two closest coordinates in a list with 1,000,000,000 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 .278 hours. Show your work (using a calculator).

- 3,200/6,400/12,800/25,600 = Ratio of Time average: ~2.25
- N = 25,600, Time Avg. = 3.49688
- 25,600 * 2^15 = 838,860,800. 3.49688 * 2.25^15 = 670,530.44401 Seconds
- 1,000,000,000 / 838,860,800 = 1.19209 / 2 = 0.59604 * 2.25 = 1.34110 Time Ratio from 838,860,800
- 670,530.44401 * 1.34110 = 899,248.37846 / 60 Seconds / 60 mins / 24 hour / 7 days = ~1.48685 weeks

Problem 2b. Answer each of the following question based on the profiles produced when running closest_2d.

1) What function/method is called the most times?

Answer: {built-in method builtins.len}, the len() function

2) (a) What function/method **called by** closest_2d (but not including itself) takes the most tottime to execute? (b) What percentage of the cumtime of closest_2d is spent in it (see your answer to part a) and the functions it calls? (c) What percentage of the total time of running this program is spent in the top 4 functions (using their tottime)? **For b-c show your calculations.**

(a) nearestneighbor.py:8(partition), the partition() function

(b) 0.195/0.752 = 0.25931 * 100 = 25.930% of cumtime

(c) 0.541/ 0.753 = 0.71846 * 100 = 71.846% of tottime

3) What is the **slowdown factor** required to execute the profiler on closest_2d: the ratio of time taken to execute the code when profiled divided by the time taken to execute the code when not profiled. Show your calculation.

Answer: 0.753 / 3.49688 = 0.21533 slowdown factor or 21.533% slowdown factor