Problem 1b. Enter 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	0.00097	No previous N to Compute Ratio
200	0.00209	2.15464
400	0.00433	2.07177
800	0.00968	2.23557
1,600	0.02231	2.30475
3,200	0.04936	2.21246
6,400	0.10750	2.17788
12,800	0.21930	2.04
25,600	0.48683	2.2099

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

Answer: O (N Log(N))

Using the last measurement in the table above, predict how long it would take to find the two closest coordinates in a list of 10^9. Express the answer in appropriate units: use larger and larger units until the number is less than 10: e.g., 1,000 seconds would be written (not as 16.666 minutes, but) as .277 hours. Show your work, using a calculator.

1,000,000,000 / 100 = 10, 000,000

10, 000,000 * Log2 (10, 000,000) = 70,000,000 * Log2 (10)

70,000,000 * Log2 (10) * 0.00097 = 225,559 s

225,559 s / 3600 s/h / 24 h/d = 2.61064 days

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

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) partition

(b) (0.266 + 0.08) / 1.126 = 30.7282%

(c) (0.443 + 0.186 + 0.09 + 0.08) / 1.126 = 70.9591%

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: 1.126 / 0.48683 = 2.31292