Question: PYTHON 3: Analysis of Algorithms/ Complexity classes PLEASE ...

PYTHON 3: Analysis of Algorithms/ Complexity classes

PLEASE ANSWER ALL PARTS SINCE THEY ARE RELATED TO EACH OTHER

In 1972, Intel's 8008 processor could execute 200,000 (200 thousand) instructions per second; at present, an Intel Core 2 processor can execute 3,200,000,000 (3.2 billion) instructions per second (16,000 times faster). Let's assume that we program the 8008 to run a fast O(N Log2 N) sorting algorithm, and program the Core 2 to run a slow O(N^2) sorting algorithm. Assume the time to sort N values on the 8008 is exactly 10/200,000 N Log2 N seconds; assume the time to sort N values on the Core 2 is 2/3,200,000,000 N^2 seconds. Here the constant for fast sorting on the 8008 is 5 times as big as the constant for slow sorting on the Core 2 (both constants are divided by the speed of the machines the algorithm runs on).

PLEASE ANSWER THE FOLLOWING QUESTIONS:
a1) About how long does it take the 8008 to sort 1,000 values?
a2) About how long does it take the Core 2 to sort 1,000 values?
b1) About how long does it take the 8008 to sort 1,000,000 values? b2) About how long does it take the Core 2 to sort 1,000,000 values?
c1) For what problem sizes N is it faster to use the Core 2 for sorting?
c2) For what problem sizes N is it faster to use the 8008 for sorting?

In problems c1 and c2 only, compute your answer to the closest integer value (you can ignore decimal places). Use a calculator, spreadsheet, or a program to compute (possibly to guess and refine) your answer.

Expert Answer

Jaqen_Hghar answered this
1,676 answers

Was this answer helpful? 0 0

Plzzzzz Upvote...
Comment for any queries thanks...

a1). As mentioned in question, the time taken to sort N values in 8008 is 100/200,000 N Log_2 N seconds

So for N = 1000, putting in above equation gives, total time = 4.98 Seconds

a2). As mentioned in question, the time taken to sort N values in Core 2 is 10/3,200,000,000 N^2 seconds

So for N = 1000, putting in above equation gives, total time = 3.125 x 10^{-3} Seconds

b1). As mentioned in question, the time taken to sort N values in 8008 is 100/200,000 N Log_2 N seconds

So for N = 1000000, putting in above equation gives, total time = 9965.78 Seconds

b2). As mentioned in question, the time taken to sort N values in Core 2 is 10/3,200,000,000 N^2 seconds

So for N = 1000000, putting in above equation gives, total time = 3125 Seconds

c1). Let for N=M, Core 2 us faster for sorting,

it means 100/200,000 M Log_2 M > 10/3,200,000,000 M^2

=> M / Log_2 M < 160000

For M <= 3519999

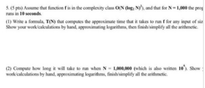
we get, M / Log_2 M < 160000, because we are considering $\lceil \log_2 M \rceil$ while calculating log. We will get Core 2 is faster.

c2). Clearly for M > 3519999 i.e. M>= 3520000 , We will get 8088 is faster.

Comment

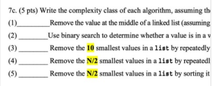
Up next for you in Computer Science

5. (5 pts) Assume that function f is in the complexity class O(N (log2 N), and that for N = 1,000 t...



See answer

7c. (5 pts) Write the complexity class of each algorithm, assuming the required data structure stor...



See answer

See more questions for subjects you study

Questions viewed by other students

Q: 1. In 1972, Intel's 8008 processor could execute 200,000 (200 thousand) instructions per second; at present, an Intel Core 2 processor can execute 3,200,000,000 (3.2 billion) instructions per second. Let's assume that we program the 8008 to run a fastO(N Log2 N) sorting algorithm, and program the Core 2 to run a slowO(N2)sorting algorithm. Assume the time to sort N values on the...

A: See answer 100% (1 rating)

Q: Fill in the last line of the three empty rows, which shows the size of a problem can be solved in the same amount of time for each complexity class on a new machine that runs nine as fast as the old one. Solve by hand when you can, use Excel or a calculator when you must: I used a calculator only for O(N Log_2 N) and solved it to 3 significant digits. Solving a problem in the same...

A: See answer 100% (5 ratings)

Show more