Question: 1. In 1972, Intel's 8008 processor could execute 200,000 (200 t…

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 8008 is 100/200,000 N Log2 N seconds; assume the time to sort N values on the Core 2 is 10/3,200,000,000 N2 seconds. Here the constant for fast sorting on the 8008 is 10 times as big as the constant for slow sorting on the Core 2 (both constants are divided by the speed of the machinesthe algorithm runs on).

a) About how long does it take the 8008 to sort 1,000 values? ~

b) About how long does it take the Core 2 to sort 1,000 values? ~

c) About how long does it take the 8008 to sort 1,000,000 values? ~

d) About how long does it take the Core 2 to sort 1,000,000 values? ~

e) For what problem sizes Nis it faster to use the Core 2 for sorting?

f) For what problem sizes N is it faster to use the 8008 for sorting?

In parts e and f 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 ⓘ

Philip answered this
1,881 answers

Was this answer helpful? 1 0

1.

a)

(100/200,000) $N\ \mathrm{Log}_2 N$ =(100/200000)*1000* $\log_2(1000)$= (1/2000)*1000* 9.97=**4.985 seconds**

b)

(10/3,200,000,000) N^2 =(10/3200000000)*(1000) ^{2}=(1/320)=**3.125 x 10^{-3}seconds**

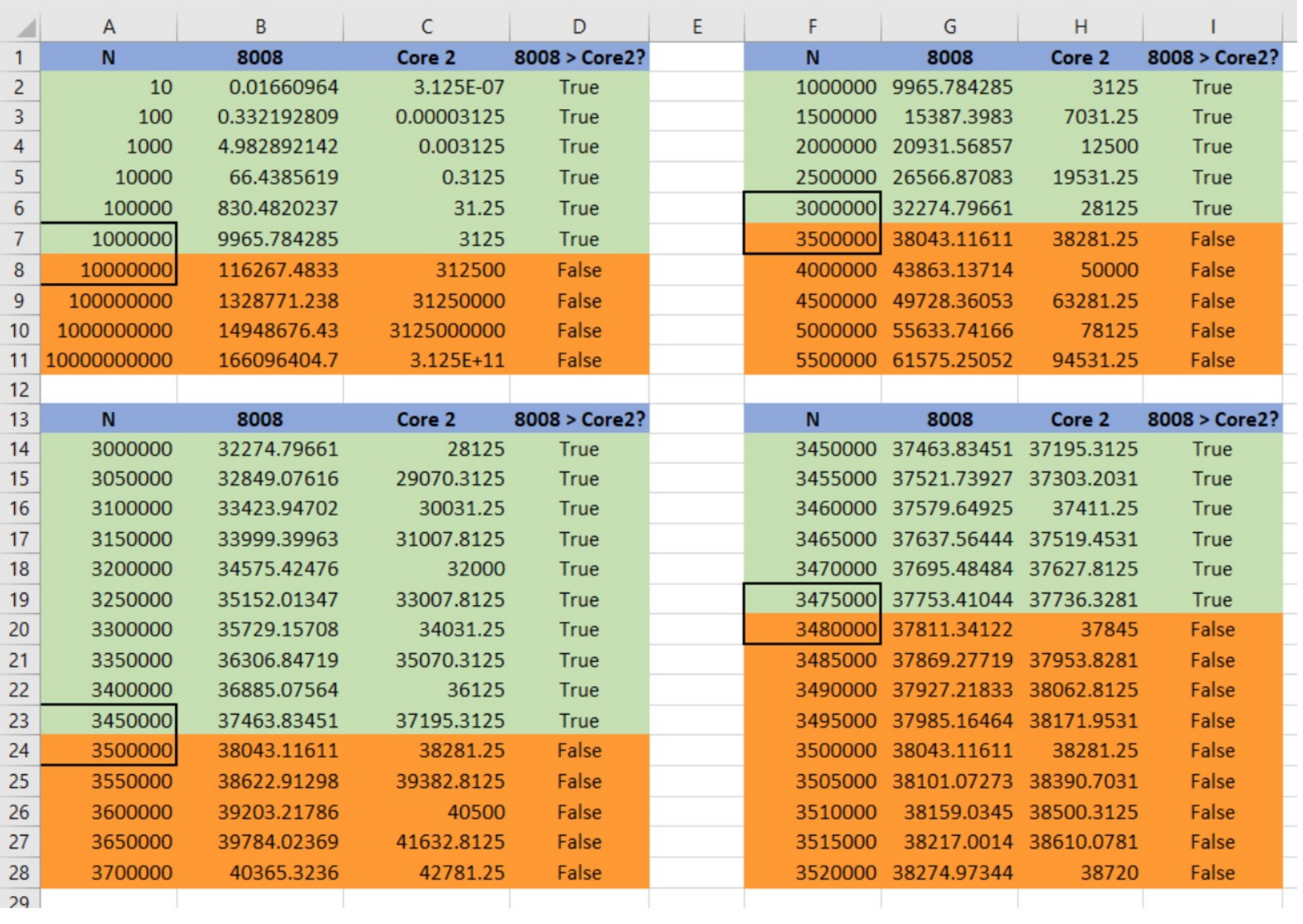
c)

(100/200,000) $N\ \mathrm{Log}_2 N$ =(100/200000)*1000000* $\log_2(1000000)$= (1/2000)*1000000* 19.93=**9965 seconds**

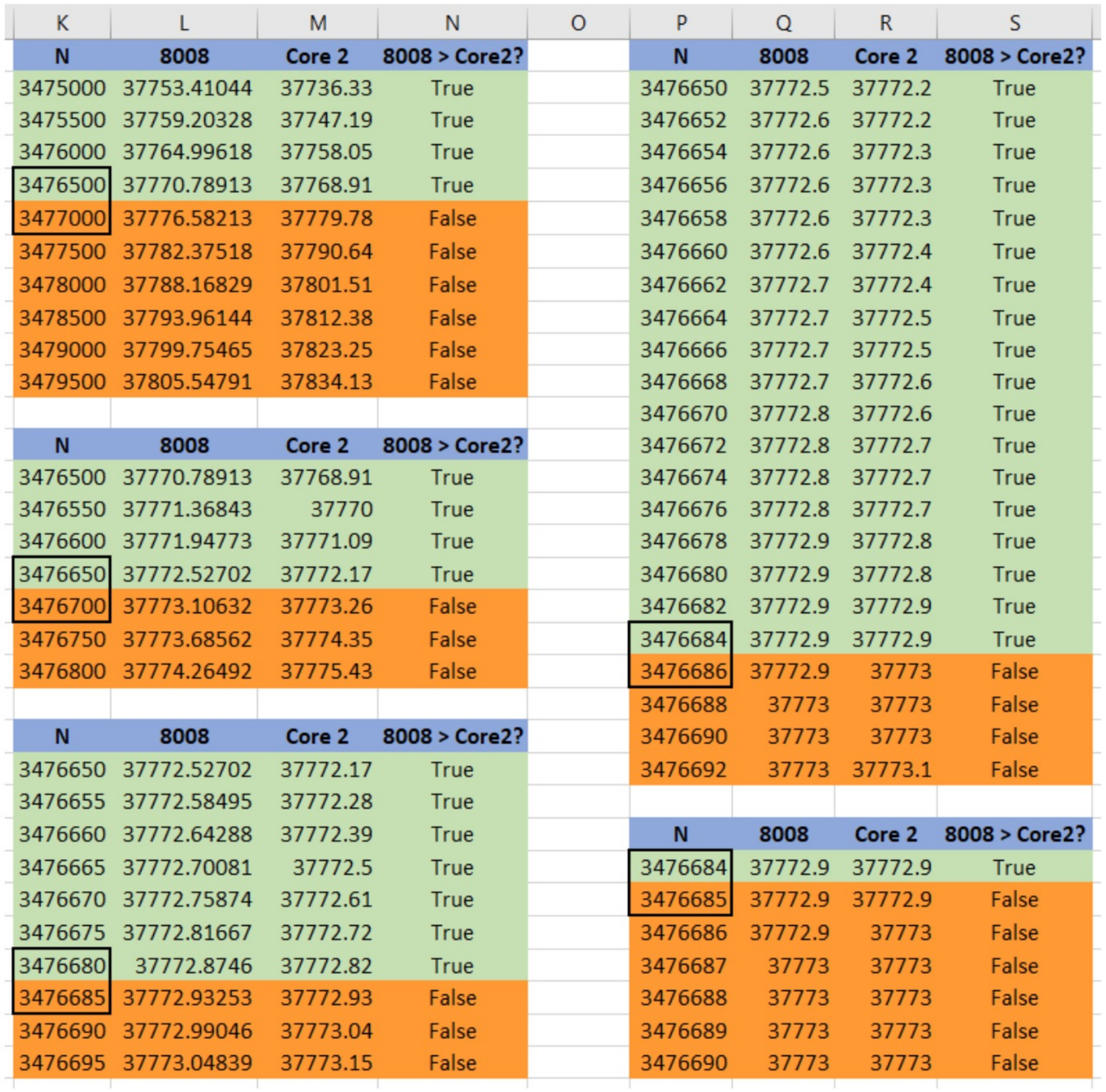
d)

(10/3,200,000,000) N^2 =(10/3200000000)*(1000000) ^{2}=(100000/32)=**3125 seconds**

Use spreadsheet (like MS-EXCEL) to find the which one is faster for different sizes as follows:



	A	B	C	D	E	F	G	H	I
1	N	8008	Core 2	8008 > Core2?		N	8008	Core 2	8008 > Core2?
2	10	0.01660964	3.125E-07	True		1000000	9965.784285	3125	True
3	100	0.332192809	0.00003125	True		1500000	15387.3983	7031.25	True
4	1000	4.982892142	0.003125	True		2000000	20931.56857	12500	True
5	10000	66.4385619	0.3125	True		2500000	26566.87083	19531.25	True
6	100000	830.4820237	31.25	True		3000000	32274.79661	28125	True
7	1000000	9965.784285	3125	True		3500000	38043.11611	38281.25	False
8	10000000	116267.4833	312500	False		4000000	43863.13714	50000	False
9	100000000	1328771.238	31250000	False		4500000	49728.36053	63281.25	False
10	1000000000	14948676.43	3125000000	False		5000000	55633.74166	78125	False
11	10000000000	166096404.7	3.125E+11	False		5500000	61575.25052	94531.25	False
12									
13	N	8008	Core 2	8008 > Core2?		N	8008	Core 2	8008 > Core2?
14	3000000	32274.79661	28125	True		3450000	37463.83451	37195.3125	True
15	3050000	32849.07616	29070.3125	True		3455000	37521.73927	37303.2031	True
16	3100000	33423.94702	30031.25	True		3460000	37579.64925	37411.25	True
17	3150000	33999.39963	31007.8125	True		3465000	37637.56444	37519.4531	True
18	3200000	34575.42476	32000	True		3470000	37695.48484	37627.8125	True
19	3250000	35152.01347	33007.8125	True		3475000	37753.41044	37736.3281	True
20	3300000	35729.15708	34031.25	True		3480000	37811.34122	37845	False
21	3350000	36306.84719	35070.3125	True		3485000	37869.27719	37953.8281	False
22	3400000	36885.07564	36125	True		3490000	37927.21833	38062.8125	False
23	3450000	37463.83451	37195.3125	True		3495000	37985.16464	38171.9531	False
24	3500000	38043.11611	38281.25	False		3500000	38043.11611	38281.25	False
25	3550000	38622.91298	39382.8125	False		3505000	38101.07273	38390.7031	False
26	3600000	39203.21786	40500	False		3510000	38159.0345	38500.3125	False
27	3650000	39784.02369	41632.8125	False		3515000	38217.0014	38610.0781	False
28	3700000	40365.3236	42781.25	False		3520000	38274.97344	38720	False
29									



K	L	M	N	O	P	Q	R	S
N	8008	Core 2	8008 > Core2?		N	8008	Core 2	8008 > Core2?
3475000	37753.41044	37736.33	True		3476650	37772.5	37772.2	True
3475500	37759.20328	37747.19	True		3476652	37772.6	37772.2	True
3476000	37764.99618	37758.05	True		3476654	37772.6	37772.3	True
3476500	37770.78913	37768.91	True		3476656	37772.6	37772.3	True
3477000	37776.58213	37779.78	False		3476658	37772.6	37772.3	True
3477500	37782.37518	37790.64	False		3476660	37772.6	37772.4	True
3478000	37788.16829	37801.51	False		3476662	37772.7	37772.4	True
3478500	37793.96144	37812.38	False		3476664	37772.7	37772.5	True
3479000	37799.75465	37823.25	False		3476666	37772.7	37772.5	True
3479500	37805.54791	37834.13	False		3476668	37772.7	37772.6	True
					3476670	37772.8	37772.6	True
N	8008	Core 2	8008 > Core2?		3476672	37772.8	37772.7	True
3476500	37770.78913	37768.91	True		3476674	37772.8	37772.7	True
3476550	37771.36843	37770	True		3476676	37772.8	37772.7	True
3476600	37771.94773	37771.09	True		3476678	37772.9	37772.8	True
3476650	37772.52702	37772.17	True		3476680	37772.9	37772.8	True
3476700	37773.10632	37773.26	False		3476682	37772.9	37772.9	True
3476750	37773.68562	37774.35	False		3476684	37772.9	37772.9	True
3476800	37774.26492	37775.43	False		3476686	37772.9	37773	False
					3476688	37773	37773	False
N	8008	Core 2	8008 > Core2?		3476690	37773	37773	False
3476650	37772.52702	37772.17	True		3476692	37773	37773.1	False
3476655	37772.58495	37772.28	True					
3476660	37772.64288	37772.39	True		N	8008	Core 2	8008 > Core2?
3476665	37772.70081	37772.5	True		3476684	37772.9	37772.9	True
3476670	37772.75874	37772.61	True		3476685	37772.9	37772.9	False
3476675	37772.81667	37772.72	True		3476686	37772.9	37773	False
3476680	37772.8746	37772.82	True		3476687	37773	37773	False
3476685	37772.93253	37772.93	False		3476688	37773	37773	False
3476690	37772.99046	37773.04	False		3476689	37773	37773	False
3476695	37773.04839	37773.15	False		3476690	37773	37773	False

e)

Therefore, for size , ***N* <= 3476684**, it is better to use Core 2 for sorting. That is, for *N*<=3476684, sorting on Core 2 is faster.

f)

Therefore, for size , ***N*>=3476685**, it is better to use 8008 for sorting. That is, for *N*>=3476685, sorting on 8008 is faster.

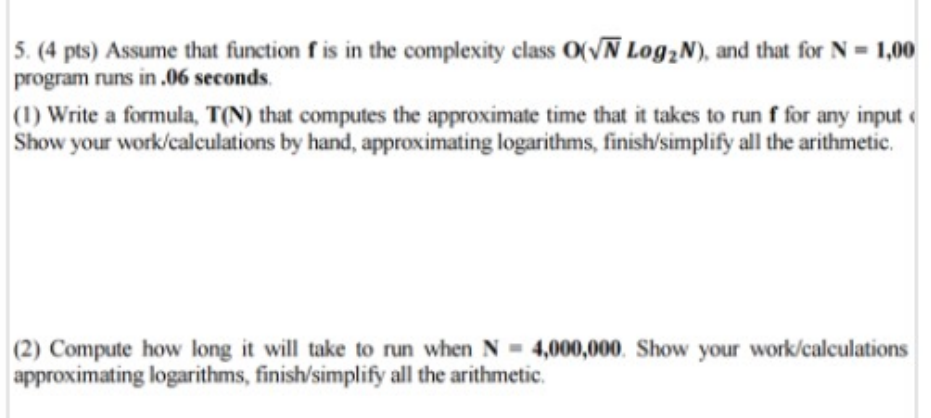
Comment ›

Up next for you in Computer Science

2. Assume that a function s is in the complexity class O(√*N*). (a) What is its doubling-signature: how much more time (by what factor) does it take to solve a problem twice as large (compute an actual number:

See answer

Assume that function f is in the complexity class O(Squareroot N Log_2 N), and that for N = 1,000,000…



5. (4 pts) Assume that function **f** is in the complexity class **O(√N Log₂N)**, and that for **N = 1,00** program runs in **.06 seconds**.

(1) Write a formula, **T(N)** that computes the approximate time that it takes to run **f** for any input Show your work/calculations by hand, approximating logarithms, finish/simplify all the arithmetic.

(2) Compute how long it will take to run when **N = 4,000,000**. Show your work/calculations approximating logarithms, finish/simplify all the arithmetic.

See answer

See more questions for subjects you study

Questions viewed by other students

Q: 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...

A: See answer

Q: The following two functions each determine the distance between the two closest values in list 1, with len(1) = N. (a) Write the complexity class of each statement in the box on its right. (b) Write the full calculation that computes the complexity class for the entire function. (c) Simplify what you wrote in (b). a = set() for i in range (len(1)): for j in range (1en(1)): if...

A: See answer 100% (2 ratings)

Show more

Post a question

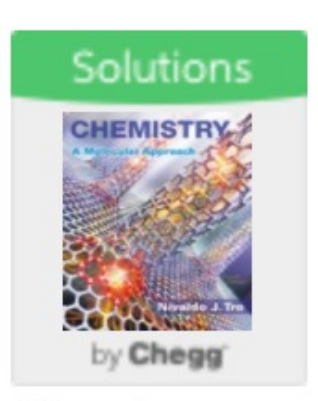
Answers from our experts for your tough homework questions

Enter question

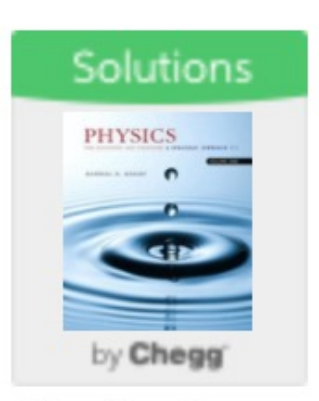
Continue to post

15 questions remaining

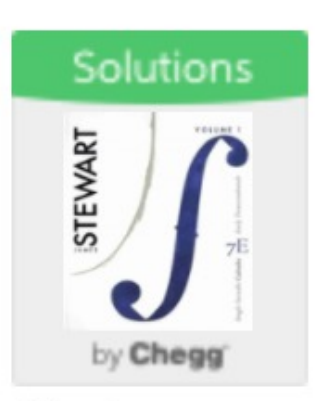
My Textbook Solutions



Chemistry
4th Edition



Physics For Scientists...
4th Edition



Single Variable...
7th Edition

View all solutions

COMPANY
About Chegg
Become a Tutor
Chegg For Good
College Marketing
Corporate Development
Investor Relations
Jobs
Join Our Affiliate Program
Media Center
Site Map

LEGAL & POLICIES
Advertising Choices
Cookie Notice
General Policies
Intellectual Property Rights
Terms of Use
Chegg Tutors Terms of Service
Global Privacy Policy
DO NOT SELL MY INFO
Honor Code

CHEGG PRODUCTS AND SERVICES
Cheap Textbooks
Chegg Coupon
Chegg Play
Chegg Study Help
College Textbooks
eTextbooks
Chegg Math Solver
Mobile Apps
Online Tutoring
Sell Textbooks
Solutions Manual
Study 101
Textbook Rental
Used Textbooks
Digital Access Codes
Chegg Money

CHEGG NETWORK
EasyBib
Internships.com
Studyblue
Thinkful

CUSTOMER SERVICE
Customer Service
Give Us Feedback
Help with Chegg Tutors
Help with eTextbooks
Help to use EasyBib Plus
Manage Chegg Study Subscription
Return Your Books
Textbook Return Policy

OVER 6 MILLION TREES PLANTED