ECE368 Non-Programming Homework Exercise #2

Due Friday, January 25, 2019, 4:30pm (MSEE 268)

IMPORTANT: Write your user (login) ID at the TOP of EACH page. Also, be sure to read and sign the Academic Honesty Statement that follows:

|  |
| --- |
| “In signing this statement, I hereby certify that the work on this exercise is my own and that I have not copied the work of any other student while completing it. I understand that, if I fail to honor this agreement, I will be subject to disciplinary action as outlined in the course policy.”  Printed Name: **Richard Rao**  Login: **rao30**  Signature:Richard Rao  I have discussed this homework with: No one |

1. For each function f(n) and time t in the following table, determine the largest size n of a problem that can be solved in time t, assuming that the algorithm to solve the problem takes f(n) microseconds. Assume that a month has 30 days and a year has 365 days.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| f(n) | t | | | | | |  |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|  | second | minute | hour | day | month | year | century |
| log2 n | 62746 | 2801417 | 133378058 | 2755147513 | 71870856404 | 797633893349 | 68610956750570 |
| Sqrt(n) | 10000 | 153261 | 2348920 | 19543806 | 188693380 | 998169750 | 21504915378 |
| n | 1000 | 7745 | 60000 | 293938 | 1609968 | 5615692 | 56156922 |
| nlog2 n | 407 | 2752 | 19109 | 87151 | 446372 | 1489517 | 13848529 |
| n2 | 100 | 391 | 1532 | 4420 | 13736 | 31593 | 146645 |
| n3 | 31 | 88 | 244 | 542 | 1268 | 2369 | 7493 |
| 2^n | 15 | 21 | 26 | 31 | 36 | 39 | 45 |
| n! | 8 | 10 | 11 | 13 | 14 | 15 | 16 |

To solve question 1,

If it takes f(n) microseconds to complete one instruction then to see how many instructions can be executed in 1 second, it will be n\*f(n)(microseconds) = 1000000(microseconds).

I have taken the floor of all values and used Stirling’s approximation for the n! problem

1. Rank the following functions by order of growth, i.e., gi = O(gi+1):

Taking n = 100

I also plugged in large numbers to verify the growth rate

* + n \*sqrt(n)

Ranked 4th because it is n^(3/2) which is of the order n^2 which is larger than n

n = 1,000

* + n!

Ranked 1st because it grows faster than ln(n)^n

N = 9.3326215443944153e157

* + (log2 n)n

Ranked 2nd because it out scales ln(n!) due to being raised to the power of n

n = 1.745911e82

* + log2(n!)

Ranked 3rd because it grows slower than n! and is not raised to the power of n

n = 273.2

* + 2^( log2n)

Ranked 5th because it is of order 2^log2(n) which equals n

n = 100

Justify your answer.

Let n be the problem size.

* 1. True or False: log2 2n = O(log2 2n/2)? Justify your answer.

True since n\*ln(2) = (n/2)\*ln(2) which are roughly the same order of magnitude

Because n = n/2\*constant; constant = 2 which is true

* 1. True or False: 2n = O(2n/2)? Justify your answer.

False since 2^(n/2)\*2^(n/2) = constant\*2^n/2

Which implies that constant = 2^n/2 which is false, hence it is not true.

* 1. True or False: 2n+1 = O(2n)? Justify your answer.

True since 2^n\*2 = 2^n\*constant

Which means constant = 2