Chapter 3: Algorithm Analysis — Questions 3-31

J.Z.W

November 10, 2023

Question 3

The number of operations executed by algorithms A and B is $40n^2$ and $2n^3$, respectively. Determine n_0 such that A is better than B for $n \geq n_0$.

Answer:

Same answer as that of Question 2. Set the two functions equal to each other and solve for n. Simplified we have $n = 20$. Thus for $n \geq 20$, A is better than B.

Question 4

Give an example of a function that is plotted the same on a *log-log* scale as it is on a standard scale.

Answer:

$$f(n) = n$$

Question 5

Explain why the plot of the function n^c is a straight line with the slope c on a *log-log* scale.

Answer:

Given a power law equation $y = an^c$, taking the log of the equation will result in $log(y) = log(an^c)$. Since a is a constant and given the logarithimic rules, $logy = clog(an)$. Thus, following the linear pattern $y = mx + b$.

Question 6

What is the sum of all even numbers from 0 to $2n$, for any positive integer n?

Answer:

$$sum = n(n + 1)$$

Question 7

Show that the following two statements are equivalent:

(a) The running time of algorithm A is always $O(f(n))$.

(b) In the worst case, the running time of algorithm A is $O(f(n))$.

Answer:

Let $A =$ the running time of algorithm A and let $W =$ the worst running time for algorithm A, be $\in A$.

If $A \implies O(f(n))$, then $\sim O(f(n)) \implies \sim A$.

Therefore, $W \implies O(f(n))$, because $W \in A$.

Question 8

Order the following functions:

$$4nlog(n) + 2n; 3n + 100log(n); n^2 + 10n; 2^{10}; 4n; n^3; 2^{log(n)}; nlog(n)$$

Answer:

$2^{10} << 4n < 3n + 100log(n) << nlog(n) < 4nlog(n) + 2n << n^2 + 10n << n^3 << 2^{log(n)} << 2^n$

Question 9

Show that if $d(n) = O(f(n)) \implies ad(n) = O(f(n))$, for any constant $a > 0$.

Answer:

Since big O notation is an approximation, the focus are on the significant growth factors. Thus, constants are usually ignored. Regardless, we know that if $d(n) = O(f(n)) \implies ad(n) = O(f(an)) = O(f(n))$, because $ad(n) \in d(n)$.

Question 10

Show that if $d(n) = O(f(n))$ and $ad(n) = O(f(n)) \implies d(n)e(n) = O(f(n)g(n))$.

Answer:

If $f(n) \cdot e(n) = n^2 \implies O(n \cdot n) = O(n^2)$ or $O(f(n)g(n))$

Question 11

Show that if $d(n) = O(f(n))$ and $e(n) \in O(g(n))) \implies d(n) + e(n) = O(f(n) + g(n))$.

Answer:

Let $d(n) = n^2$ and $e(n) = n$

$O(d(n)) = n^2$ and $O(e(n)) = n$ $d(n) + e(n) = n^2 + n \implies O(n^2 + n) = n^2 = O(n^2) + O(n)$

Question 12

Show that if $d(n) = O(f(n))$ and $e(n) \in O(g(n))) \implies d(n) - e(n)$ is not necessarily $O(f(n) - g(n))$.

Answer:

Let $d(n) = 2n$ and $e(n) = n$

If $d(n) - e(n) = n \implies O(d(n) - e(n)) = O(n)$
If $d(n) = O(n)$ and $e(n) = O(n) \implies O(d(n) - e(n)) = O(1) \neq O(d(n) - e(n))$

Question 13

Show that if $d(n) = O(f(n))$ and $f(n) \in O(g(n))) \implies d(n) = O(g(n)$

Answer:

Let $g(n) = n$ and $f(n) = g(n)$

If $d(n) = O(f(n))$ and $O(f(n)) = O(g(n)) \implies d(n) = O(g(n))$

Question 14

Show that if $O(max\{f(n), g(n)\}) = O(f(n) + g(n))$

Answer:

Let $g(n) <<< f(n)$

$max\{g(n), f(n)\} = f(n) \implies O(max\{f(n), g(n)\}) = O(f(n)) + O(g(n)) = O(f(n))$

Question 15

Show that if $f(n) = O(g(n)) \iff g(n) = \Omega(f(n))$

Answer:

Firstly we must know that Big-Ω is defined as being asymptotically less than or equal to said function. Therefore if $f(n) = O(g(n))$, we know that the estimated time complexity is $f(n) = g(n)$, thus satisfying one of the definitions of Big-Ω

Question 16

Show that if $p(n) \in n \in \mathbb{P} \implies log(p(n)) = O(log(n))$

Answer:

Since we know $p(n)$ is equivalent to n, then $logp(n) = logn$. Therefore $logp(n) = O(log(n))$ since $log(n) = O(log(n))$

Question 17

Show that $(n+1)^5 = O(n^5)$

Answer:

By defintion of Big-O, the highest polynomial would be cn^5, thus being $O(n^5)$ time complexity

Question 18

Show that $2^{n+1} = O(2^n)$

Answer:

The addition of the constant in the exponent is negligible.

Question 19

Show that $n = O(nlog(n))$

Answer:

We know that big O is defined as the worst case, so even though it is more accurate to say $n = O(n)$, it is still true to say $n = O(nlogn)$ because $n << nlog(n)$

Question 20

Show that $n^2 = \Omega(nlog(n))$

Answer:

By definition of Big-Omega, $f(x) = \Omega(g(x))||g(x))$ is asymptotically same or lower as $(fx)||f(x) >= O(g(x))$

Question 21

Same as the above

Question 22

Show that $f(n) = O(f(n))$, if $f(n)$ is a positive nondecreasing function that is always greater than 1.

Answer:

Definition of big-O, given the fact that $f(n)$ is a positive nondecreasing function

Question 23

Give the big-Oh characterization in terms of n for the running time of the code fragment 3.23.

Answer:

$$O(n)$$

Question 24

Give the big-Oh characterization in terms of n for the running time of the code fragment 3.24.

Answer:

$$O(log(n))$$

Question 25

Give the big-Oh characterization in terms of n for the running time of the code fragment 3.25.

Answer:

$$O(n^2)$$

Question 26

Give the big-Oh characterization in terms of n for the running time of the code fragment 3.26.

Answer:

$$O(n)$$

Question 27

Give the big-Oh characterization in terms of n for the running time of the code fragment 3.27.

Answer:

$$O(n^3)$$

Question 28

Calculate the largest size of n within a given time for each $f(n)$

Answer:

Simply use the first value of $f(n)$ and extrapolate

Question 29

Algorithm A executes at $O(log(n))$ time complexity for each entry of an n-element sequence. What is its worst-case running time?

Answer:

$O(nlog(n))$ because it iterates through n and each element in n takes $O(log(n))$

Question 30

Given a n-elment sequence S, Algorithm B chooses $log(n)$ elements in S at random and executes an $O(n)$ time calculation for each. What is the worst-case running time for Algo B?

Answer:

$O(nlog(n))$

Question 31

Given a n-elment sequence S of integers, Algorithm C executes an $O(n)$-time calculation for each even number in S, and an $O(log(n))$-time computation for each odd number in S. What are the best-case and worst-case running times of Algo C?

Answer:

For the best case, there are no odd numbers and thus $O(n)$, but if there are all odd numbers then $O(nlog(n))$ would be the worst case.