Noah Hayek (27080409) | Assignment 1 - COMP 352

Sun, May 15, 2016 12:48 AM

viii) f(n) = 2^n; g(n) = $10n^2$.

Ans:

$$O(f(n)) = O(2^n) \; and \; O(g(n)) = O(n^2) \Rightarrow \Omega(f(n)) = \Omega(2^n) \; and \; \Omega(g(n)) = \Omega(n^2).$$
$$f(n) \; is \; in \; O(g(n)) \; since \; O(n^2) \subset O(2^n). \; But \; f(n) \; is \; not \; in \; \Omega(g(n)) \; since \Omega(n^2) \not\subset \Omega(2^n)$$
$$and \; based \; on \; that, \; f(n) \; is \; not \; in \; \Theta(g(n)).$$

ix) f(n) = 2^n; g(n) = n log n.

Ans:

$$O(f(n)) = O(n^2) \; and \; O(g(n)) = O(n \; log(n)) \Rightarrow \Omega(f(n)) = \Omega(n^2) \; and \; \Omega(g(n)) = \Omega(n \; log(n))$$
$$f(n) \; is \; in \; O(g(n)) \; since \; O(n \; log(n)) \subset O(n^2).$$
$$But \; f(n) \; is \; not \; in \; \Omega(g(n)) \; since \Omega(n \; log(n)) \not\subset \Omega(n^2)$$
$$and \; based \; on \; that, \; f(n) \; is \; not \; in \; \Theta(g(n)).$$

x) f(n) = 2^n; g(n) = 3^n.

Ans:

$$O(f(n)) = O(2^n) \; and \; O(g(n)) = O(3^n) \Rightarrow \Omega(f(n)) = \Omega(2^n) \; and \; \Omega(g(n)) = \Omega(3^n).$$
$$2^n \; and \; 3^n \; are \; two \; exponential \; functions \; with \; different \; base \; numbers \; and \; therefore,$$
$$are \; not \; of \; the \; same \; order \; since \; 2^n < 3^n \; for \; n > 0.$$
$$Based \; on \; that \; O(3^n) \not\subset O(2^n) \; and \; f(n) \; is \; not \; in \; O(g(n)) \; and \; is \; not \; in \; \Theta(g(n)).$$
$$On \; the \; other \; hand, \; \Omega(3^n) \subset \Omega(2^n) \; since \; the \; \Omega \; hierarchy \; of \; elements \; is \; the \; reverse$$
$$of \; that \; of \; Big \; O.$$
$$\therefore f(n) \; is \; in \; \Omega(g(n)).$$

xi) f(n) = 2^n; g(n) = n^n.

Ans:

$$O(f(n)) = O(2^n) \; and \; O(g(n)) = O(n^n) \Rightarrow \Omega(f(n)) = \Omega(2^n) \; and \; \Omega(g(n)) = \Omega(n^n).$$
$$f(n) \; is \; not \; in \; O(g(n)) \; because \; n^n \; is \; of \; higher \; order \; than \; 2^n.$$
$$This \; also \; indicates \; that \; f(n) \; is \; not \; in \; \Theta(g(n)).$$
$$However, \; since \; the \; hierarchy \; of \; elements \; of \; \Omega \; is \; the \; inverse \; of \; that \; of \; Big \; O,$$
$$\Omega(2^n) \; is \; in \; \Omega(n^n) \; since \; \Omega \; (n^n) \subset \Omega(2^n).$$

b) The following run times were obtained when using two different algorithms on a data set of size *n*. Based on this timing data, guess at the asymptotic complexity of each algorithm as a function of *n*. Use Big-O notation in its simplest form and briefly explain how you reached your conclusion.

i)

n	Execution Time (in seconds)
1000	0.743
2000	3.021
4000	12.184
8000	50.320

Ans: O(n), since the execution time increases by a factor of approximately 4 when the value of n is doubled.

ii)

n	Execution Time (in microseconds or millionths of a second)
1000	0.01
1000000	20
1000000000	30000

Ans: O(log(n)), since the execution prove that the algorithm is effective.