Prove that the running time of an algorithm is $\Theta(g(n))$ if and only if its worst-case running time is $O(g(n))$ and its best-case running time is $\Omega(g(n))$.

A: ∵ for the running time to be $\Theta(g(n))$ the $f(n)$ must satisfy :

$$c_1 . g_1(n) \geq f(n) \geq c_2 . g_2(n) \qquad (1)$$

∵ at the worst-case the $f(n)$ satisfy :

$$f(n) \leq c_1 . g_1(n) \qquad \text{Where } f(n) = O(g(n)) \quad (2)$$

∵ at the best-case the $f(n)$ satisfy :

$$f(n) \geq c_2 . g_2(n) \qquad \text{Where } f(n) = \Omega(g(n)) \quad (3)$$

from (1), (2), (3) :-

∴ the running time of an algorithm is $\Theta(g(n))$ if and only if its worst-case running time is $O(g(n))$ and its best-case running time is $\Omega(g(n))$ #
