The Design and Analysis of Algorithms's homework

BeanCb

28th September 2016

3.1-2 Show that for any real constants a and b, where $b > 0$,

$$(n+a)^b = \Theta(n^b) \tag{3.2}$$

Answer:

To show $(n+a)^b = \Theta(n^b)$, first we find constants $c1$, $c2$, $n_0 > 0$ so we can get :

$$0 \leqslant c_1 n^b \leqslant (n+a)^b \leqslant c_2 n^b$$

for all $n \leqslant n_0$.

Note that

$$n + a \leqslant n + |a| \leqslant 2n \ when \ |a| \leqslant n \tag{3.2-1}$$

and

$$n + a \geqslant n - |a| \geqslant \frac{1}{2} n \ when \ |a| \leqslant \frac{1}{2} n \tag{3.2-2}$$

when $n \geqslant 2|a|$,

$$0 \leqslant \frac{1}{2} n \leqslant n + a \leqslant 2n. \tag{3.2-3}$$

Since $b > 0$, it will still holds when all parts are raised to the b:

$$0 \leqslant (\frac{1}{2} n)^b \leqslant (n+a)^b \leqslant (2n)^b, \tag{3.2-4}$$

$$0 \leqslant (\frac{1}{2})^b n^b \leqslant (n+a)^b \leqslant 2^b n^b. \tag{3.2-5}$$

So $c_1 = (\frac{1}{2})^b$, $c_2 = 2^b$, and $n_0 = 2|a|$ satisfy the difinition.

3.1-3 Explain why the statement,"The running time of algorithm A is at least $O(n^2)$," is meaningless.

Answer:

The running time of algorithm A is $T(n)$. $T(n) \geqslant O(n^2)$ means $T(n) \geqslant f(n)$ for some function $f(n)$ in the set $O(n^2)$.We get an upper bound for the worst situation to be the lower bound of the algorithm.So we know nothing about the running time.

3.1-4 Is $2^{n+1} = O(2^n)$? Is $2^{2n} = O(2^n)$?

Answer:

$$2^{n+1} = O(2^n),\ 2^{2n} \neq O(2^n) \tag{3.4-1}$$

To show $2^{n+1} = O(2^n)$,we must find constant $c,\ n_0 > 0$ so we can get

$$0 \leqslant 2^{n+1} \leqslant c \cdot 2^n\ for\ all\ n \geqslant n_0. \tag{3.4-2}$$

Both side divide 2^n so we can get

$$0 \leqslant 2 \leqslant c \tag{3.4-3}$$

So we can satisfy the definition with $c \geqslant 2$ and $n_0 \geqslant 1$.

To show $2^{2n} \neq O(2^n)$, we assume there exist constants $c,\ n_0 > 0$ so we can get

$$0 \leqslant 2^{2n} \leqslant c \cdot 2^n\ for\ all\ n \leqslant n_0 \tag{3.4-4}$$

Then both side divide 2^n so we can get

$$0 \leqslant 2^n \leqslant c \tag{3.4-5}$$

So we can get $c \geqslant 2^n$ to satisfy the definition, but no constant is greater than all 2^n,so the assumption does't hold.