Solutions

2.3-3

Use mathematical induction to show that the solution to the recurrence:

$$T(n) = \begin{cases} 2 & \text{if } n = 2 \\ 2T(\frac{n}{2}) + n & n = 2^k, k > 1. \end{cases}$$

is $T(n) = n \lg n$.

Solution

Proof.
Step 1: Base Case

Verify the base case $T(n) = n \log n$ for $n = 2$
$T(2) = 2 \lg 2$
$T(2) = 2$
Thus , $T(2) = 2 \log 2$

Step 2: Inductive Assumption

Let us assume $T(n) = n \log n$ is true for for $n = 2^k$
i.e $T(2^k) = 2^k \log 2^k$ holds true.

Step 3: Induction Step

If $n = 2^{k+1}$
$T(2^{k+1}) = 2T(\frac{2^{k+1}}{2}) + 2^{k+1}$
$= 2T(\frac{2^k 2}{2}) + 2^{k+1}$
$= 2T(2^k) + 2^{k+1}$
$= 2(2^k \log 2^k) + 2^{k+1}$
$= 2^{k+1} \log 2^k + 2^{k+1}$
$= 2^{k+1}((\log 2^k) + 1)$
$= 2^{k+1} \log 2^{k+1}$
Thus , the result holds for all $k > 1$.

$\square$

2-4

(a) List the five inversions of the array $\langle 2, 3, 8, 6, 1 \rangle$

(b) What array with elements from the set $\{1, 2, ..., n\}$ has the most inversions? How many does it have?

(c) What is the relationship between the number of inversions in an array and the running time of insertion sort? Justfy your answer.

Solution

(a) (1,5),(2,5),(3,4),(3,5),(4,5).

(b) The array will have maximum inversions when the elements of array are in descending order.
Array : [n,n-1,n-2,.....,3,2,1]
Total no. of inversions = $(n-1) + (n-2) + ... + 2 + 1$

$$\sum_{1}^{n-1} k = \frac{n(n-1)}{2}$$

(c) The running time of insertion sort is directly proportional to the number of inversions in an array. According to the insertion sort algorithm , the logic inside while loop causes inversions. More the number of inversions in the array,the more will be number of times the while loop is executed which will increase the running time of insertion sort.

3.1-2

Show that for any real constants a and b, where $b > 0$, $(n + a)^b = \Theta(n^b)$

Solution

We can expand the expression $(n + a)^b$ using binomial expansion.

$$\begin{aligned}(n + a)^b &= \frac{b!}{0!b!}n^b a^0 + \frac{b!}{1!(b-1)!}n^{b-1}a^1 + ... + \frac{b!}{(b-1)!1!}n^1 a^{b-1} + \frac{b!}{b!0!}n^0 a^b \\ &= c_1 n^b + c_2 n^{b-1} + ... + c_{z-1} n^1 + c_z n^0\end{aligned}$$

We can see that the highest order term after binomial expansion is n^b. By throwing away the lower-order terms and ignoring the leading coefficients of the highest order term , we can conclude that $(n + a)^b = \Theta(n^b)$.

3.1-4

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

Solution

We know that ,
$O(g(n)) = \{\ f(n)$: there exist positive constants c and n_0 such that
$0 \leqslant f(n) \leqslant cg(n)$ for all $n \geqslant n_0\ \}$

Therefore,
$f(n) = 2^{n+1} = 2(2^n)$
Let $n_0 = 1$ and c=3 then
$0 \leqslant 2^{1+1} \leqslant 3(2^1)$
i.e $0 \leqslant 4 \leqslant 6$
Hence , $2^{n+1} = O(2^n)$

Now , $f(n) = 2^{2n}$
$= 2^{n+n} = (2^n)(2^n)$
$(2^n)(2^n) \leqslant c(2^n)$
So , $2^n \leqslant c_1$
This cannot be true because n is a variable.
Hence , $2^{2n} = O(2^n)$ is false

3-2

Indicate for each pair of expressions (A, B) whether A is big O, little o, big Ω, little *omega*, or Θ of B. The expressions involve some constants; assume $k >= 1$, $\epsilon > 0$, and $c > 1$ are constants. Your answer should be in the form of a table with "yes" or "no" written in each box.

	A	B	O	o	Ω	ω	Θ
a.	$\lg^k n$	n^ϵ					
b.	n^k	c^n					
c.	$\sqrt{n}$	$n^{\sin n}$					
d.	2^n	$2^{n/2}$					
e.	$n^{\lg c}$	$c^{\lg n}$					
f.	$\lg n!$	$\lg n^n$					

Solution

	A	B	O	o	Ω	ω	Θ
a.	$\lg^k n$	n^ϵ	Yes	Yes	No	No	No
b.	n^k	c^n	Yes	Yes	No	No	No
c.	$\sqrt{n}$	$n^{\sin n}$	No	No	No	No	No
d.	2^n	$2^{n/2}$	No	No	Yes	Yes	No
e.	$n^{\lg c}$	$c^{\lg n}$	Yes	No	Yes	No	Yes
f.	$\lg n!$	$\lg n^n$	Yes	No	Yes	No	Yes

A.1-1

Find a simple formula for

$$\sum_{k=1}^{n}(2k-1)$$

Solution

We can simplify the formula using linearity property

$$\begin{aligned}
\sum_{k=1}^{n}(2k-1) &= 2\sum_{k=1}^{n}(2k) - \sum_{k=1}^{n}(1) \\
&= 2(\frac{n(n+1)}{2}) - n \\
&= n^2 + n - n \\
&= n^2
\end{aligned}$$