التاريخ:

محمد اسامة محمد حلمي عبد الهادي

التاريخ:

الموضوع:

H.W 2

order the following expressions -1

$2^{\log_2 n} < n \log n < n^3 < 2^n < n! < 2^n < e^n < n! < n^{2n}$

$(\log n! < n^{\log \log n})$

$f(n) = \frac{n^3}{1000} - 100n^2 - 100n + 3$ -2

1- identify the dominant term

The growth of each term is based on its degree of n n

$\frac{n^3}{1000}$ cubic term (n^3)

$-100n^2$ quadratic term (n^2)

$-100n$: linear term (n^1)

3 Constant term

2- ignore coefficient and lower order terms

in Big-o notation, we are interested in the order of growth, not the exact coeffecients. So $\frac{n^3}{1000}$ simplifies to n^3

other terms ($-100n^2, -100n, 3$) are ignored

التاريخ: / /

الموضوع:

find Big-O expression
The function grows asymptotically like n^3 Therefore: $fn \in O(n)^3$

Explanation: The dominant term n^3 dictates the growth rate as n increases and all other terms contribute insignificantly to the overall complexity for large n Thus, the function is expressed as $O(n^3)$

3 الصفحة رقم

To analyze the recursive Version of the Insertion sort algorithm, we will

1- Define the problem 2- write the recurrence relation
3- solve the recurrence using the master Theroem or substitution method

Recursive Algorithm for insertion sort.

The recursive algorithm Works as follows

1- if the array A has only one element no 1 it is already sorted Basecase
$T(1) = \theta(1)$ 2 for $n > 1$

التاريخ: / /

الموضوع:

Recursively sort the first n-1 elements ($A[1 \ldots n-1]$)

Insert $A[n]$ into the sorted subarray $A[1 \ldots n-1]$

Recurrence Relation

The Recurrence for the running time $T(n)$ is

$T(n) = T(n-1) + \theta(n)$

$T(n-1)$: Time to sort the first n-1 elements recursively

$\theta(n)$ Time to insert $A[n]$ into the sorted subarray, which involves a linear scan

---

Solving the recurrence

1- using substitution

Expand the recurrence

$T(n) = T(n-1) + \theta(n)$

$T(n) = T(n-2) + \theta(n-1) + \theta(n)$

$T(n) = T(n-3) + \theta(n-2) + \theta(n-1) + \theta(n)$

Continue expanding until $T(1)$

$$T(n) = T(1) + \sum_{k=2}^{n} \theta(k)$$

since $T(1) = \theta(1)$ the total running time is

التاريخ: / / الموضوع:

$$T(n) = \theta(1) + \sum_{k=2}^{n} \theta(k)$$

The summation $\sum_{k=2}^{n} k$ is equivalent to

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

So the running time simplifies to

$$T(n) = \theta\left(\frac{n^2}{2}\right) = \theta(n^2) \quad \ldots (1)$$

2 using master Therom

The recurrence is

$$T(n) = T(n-1) + \theta(n)$$

Final Running Time

The running time of recursive insertion is

$$T(n) = \theta(n^2)$$

التاريخ: / / الحاسبة

المسألة الرابعة

الموضوع: التاريخ: / /

$$T(n) = \begin{cases} 1 & \text{if } n = 1 \\ T(n-1) + n(n-1) & \text{if } n \geq 2 \end{cases}$$

step 1

$$T(n) = T(n-1) + n(n-1)$$

substitute $T(n-1)$

$$T(n) = (T(n-2) + (n-1)(n-2)) + n(n-1)$$

$$T(n) = T(n-2) + (n-1)(n-2) + n(n-1)$$

substitute $T(n-2)$

$$T(n) = (T(n-3) + (n-2)(n-3)) + (n-1)(n-2) + n(n-1)$$

$$T(n) = T(n-3) + (n-2)(n-3) + (n-1)(n-2) + n(n-1)$$

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

التاريخ: / /

الموضوع:

Simplify the summation

The summation term is

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

simplify $k(k-1)$ as $k(k-1) = k^2 - k$

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

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

$$\sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}$$

$$\sum_{k=2}^{n} k^2 = \sum_{k=1}^{n} k^2 - 1^2 = \frac{n(n+1)(2n+1)}{6} - 1$$

Compute $\sum_{k=2}^{n} k$ $\quad \sum_{k=1}^{n} k = \frac{n(n+1)}{2}$

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

3 Combine the results

التاريخ: / / الموضوع:

$$\sum_{k=2}^{n} k(k-1) = \left(\frac{n(n+1)(2n+1)}{6} - 1\right)\left(\frac{n(n+1)}{2} - 1\right)$$

simplify $\sum_{k=2}^{n} k(k-1) = \frac{n(n+1)(2n+1)}{6} - \frac{n(n+1)}{2}$

$$\sum_{k=2}^{n} k(k-1) = \frac{n(n+1)}{6}(2n+1-3)$$

$$\sum_{k=2}^{n} k(k-1) = \frac{n(n+1)}{6}(2n-2)$$

$$\sum_{k=2}^{n} k(k-1) = \frac{n(n+1)(n-1)}{3}$$

Total running time

The total running time

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

$$T(n) = 1 + \frac{n(n+1)(n-1)}{3}$$

$$\therefore T(n) = \theta\left(\frac{n^3}{3}\right) = \theta(n^3)$$

المسألة الخامسة

التاريخ: / / الموضوع:

$T(n) = 7T(\frac{n}{2}) + n^2$

step 1 Expand the recurrence

$T(n) = 7T\frac{n}{2} + n^2$

substitute $T(\frac{n}{2})$

$T(n) = 7[7T(\frac{n}{4}) + (\frac{n}{2})^2] + n^2$

$T(n) = 7^2 T(\frac{n}{4}) + 7(\frac{n}{2})^2 + n^2$

substitute $T(\frac{n}{4})$ $T(n) = 7^2[7T(\frac{n}{8} + n^2)]$
$+ 7(\frac{n}{2})^2 + n^2$

$T(n) = 7^3 T(\frac{n}{8}) + 7^2(\frac{n}{4})^2 + 7(\frac{n}{2})^2 + n^2$

Generalize the pattern

$T(n) = 7^k T(\frac{n}{2^k}) + \sum_{i=0}^{k-1} 7^i (\frac{n}{2^i})^2$

Base case
When $k = \log_2 n$ the problem size becomes $T(1)$ and we stop the recursion $T(1) = \Theta(1)$
substitute $k = \log_2 n$ into the generalized form

$T(n) = 7^{\log_2 n} T(1) + \sum_{i=0}^{\log_2 n - 1} 7^i (\frac{n}{2^i})^2$

التاريخ: / /

الموضوع:

$$a^{\log_b x} = x^{\log_b a}$$

$$7^{\log_2 n} = n^{\log_2 7}$$

$$7T\left(\frac{n}{2}\right) + T(n) = \Theta(n^{\log_2 7})$$

$$\sum_{i=0}^{\log_2 n - 1} 7^i \left(\frac{n}{2^i}\right)^2 : \text{Expand } \left(\frac{n}{2^i}\right)^2 = \frac{n^2}{4^i}$$

$$\sum_{i=0}^{\log_2 n - 1} 7^i \left(\frac{n}{2^i}\right)^2 = n^2 \sum_{i=0}^{\log_2 n - 1} \frac{7^i}{4^i}$$

$$\frac{7^i}{4^i} = \left(\frac{7}{4}\right)^i \quad \sum_{i=0}^{\log_2 n - 1} \frac{7^i}{4^i} = \sum_{i=0}^{\log_2 n - 1} \left(\frac{7}{4}\right)^i$$

for large n the dominant term is r^k

$$\sum_{i=0}^{\log_2 n - 1} \left(\frac{7}{4}\right)^i \approx \left(\frac{7}{4}\right)^{\log n}$$

using $a^{\log_b x} = x^{\log_b a}$ $\left(\frac{7}{4}\right)^{\log_2 n} = n^{\log_2 \left(\frac{7}{4}\right)}$

so the summation becomes

$$n^2 \cdot n^{\log_2 (7/4)} = n^{2 + \log_2 (7/4)}$$

combine results
the total running time

التاريخ: / /

الموضوع:

$T(n) = \theta(n^{\log_2 7}) + \theta(n^2 + \log_2(7/4))$

$\therefore T(n) = \theta(n^{\log_2 7})$

$\because \log_2 7 \simeq 2{,}8$ and $2 + \log_2(7/4) \simeq 2{,}8$

The final Result is

$T(n) = \theta(n^{\log_2 7})$.

المسألة الاضافية في واجب (2)

التاريخ: / /

الموضوع:

$$T(n) = 8T\left(\frac{n}{2}\right) + n^2$$

Recall Therom

$$T(n) = aT\left(\frac{n}{b}\right) + f(n)$$

$a \geq 8$ $\quad$ $b \geq$ The factor by which the input size is divided

$f(n) = n^2$; The cost out size the recursive call

$p = \log_b a \geq 2$ where $p = \log_2 8 = 3$

compare $f(n) = n^2$ with $n^p = n^3$

---

compare $f(n)$ and n^p

$f(n) = n^2$ grows slower than n^3

By master theorem if $f(n) = O(n^p / \log^* n)$

the solution to the recurrence is dominated by n^p

Thus since $f(n) = O(n^p)$ with $p = 3$ the asymptotic is

$T(n) = \Theta(n^3)$

⇒ final Result is $T(n) = \Theta(n^3)$