Tutorial 2

Question 1

A sorting method with "Big-Oh" complexity $O(n.\log_{10}n)$ spends exactly 1 millisecond to sort 1,000 items. Assuming that time T(n) of sorting n items is directly proportional to $n.\log_{10}n$, that is, $T(n)=(c.n.\log_{10}n)$, where c is a constant. Derive a formula for T(n), given the time T(x) for sorting x items, and estimate how long this method will sort 1,000,000 items.

Answer:

Processing time is $T(n)=c.n.\log_{10}n$

$T(n) / (n.\log_{10}n) = c$

Given T(x) is the time to process x items

$C = T(x) / (x. \log_{10}x)$

Hence

$T(n) = (T(x) / (x. \log_{10}x)) . n . \log_{10}n$

Given T(1000) = 1 msec is the time to process 1000 items

$T(n) = (T(1000) / (1000*\log_{10}1000)) . n . \log_{10}n$

taking log base 10

$T(n)=(T(1000) / (1000*\log_{10}10^3)) . n . \log_{10}n$

$T(n)= (T(1000) / 3000) . n . \log_{10}n$

This spends 1 millisecond to sort 1000 items. Hence T(1000)=1ms

$T(n) = n . \log_{10}n / 3000$ ms

T(1000000) = 1000000log(1000000)/3000 ms

T(1000000) = 6000000/3000 ms

T(1000000) = 2000 ms

Question 2

Assume that each of the expressions below gives the processing time T(n) spent by an algorithm for solving a problem of size n. Select the dominant term(s) having the steepest increase in n and specify the lowest Big-Oh complexity of each algorithm.

Expression	Dominant term(s)	Big-Oh complexity
$5+0.0001n^3+0.025n$	$0.0001n^3$	$O(n^3)$
$500n+100n^{1.5} + 50n\log_{10}n$	$100n^{1.5}$	$O(n^{1.5})$
$n^2\log_2n+n(\log_2n)^2$	$n^2\log_2n$	$O(n^2\log_2n)$

Tutorial 2

Question 3:

The number of operations executed by algorithms A and B is $8n \lg n$ and $2n^2$, respectively. Determine n^0 such that A is better than B for $n \geq n^0$.

Answer:

Setting two equations equal and simplifying;

$2n^2 = 8n \lg n$

$n = 4 \lg n$

we see that the cross-over occurs for $n = 16$. Since the logarithmic function grows much slower than the quadratic function, for all $n \geq 16$ we will have $8n \lg n \leq 2n^2$, i.e., A is better than B.