Problem 1: <u>True</u> If we consider sorting of n number of elements in an array, then

a[n] of length n. If we consider comparison based algorithms as an associated decision tree or a Binary Search tree then we can see that the Binary tree has at least $n!$ leaves and the depth is $\Omega(n \log(n))$

then if we consider. $|S| = n!$ then splitting S into 2 groups, YES and NO and the adversary gives each comparison on the larger value then S will cut down by size at most factor of 2 So,

Algorithm must take at least $\log_2 n!$ comparisons

According to <u>Stirling formula</u>,

$$n! = 2^{n \log n}$$

$$\boxed{\log n! \; n \log n}$$

$$\log n! = \log(n(n-1)(n-2)\ldots(2))$$

$$= \sum_{i=2}^{n} \log i = \Omega(n \log n)$$

$\Omega(n \log n)$ is the lower bound

Scanned with CamScanner

Broblem 2 :

As we know, $o(g(n))$ means that there are functions $g(n)$ or $f(n)$

such that $0 \le f(n) \le c_1 g(n)$ for any positive constant $c_1 > 0$ and $n \ge n_0$

and $\omega(g(n))$ is the function $g(n)$ such that

$$0 \le c_2 g(n) < f(n) \text{ for any } c_2 > 0$$

so, the intersection of function is such that

$0 \le c_2 g(n) < f(n) < c_1 g(n)$ so as n becomes large and $f(n)$ cannot be greater than $c_2 g(n)$ and less than $c_1 g(n)$

so, no $f(n)$ exists

$$o(g(n)) = \lim_{n \to \infty} \frac{f(n)}{g(n)} = 0 \qquad \omega(g(n)) = \lim_{n \to \infty} \frac{f(n)}{g(n)} = \infty$$

Cannot be true hence $f(n)$ does not exist.

Problem 3.

1. As

$f = \Theta(g)$

If $c_1 = 1$ and $c_2 = 200$ $n \geq 0$

$0 \leq g(n) \leq f(n \leq 200\, g(n)$

$f = \Theta(g)$

2.

$f =$ Big omega$(g(n))$

For large value of n

$0 \leq g(n) \leq f(n)$ $f =$ Bigomega$(g(n))$

3. f is Bigomega.$(g(n)$

So, $n \geq 0$, $\log_2 n > \log_3 n$

4. $f =$ Big $O(g)$

$2n \leq 3n$

5. $f =$ Big $O(g(n)$

$0.5n \leq 1$, $n \leq 1$

$f =$ Big $O(g)$

Problem 4 :

a) As we know, $f(n) = O(g(n))$ such that there exist +ve constants c and n_0

$g(n) = O(f(n))$ $\qquad f(n) \le c \times g(n) \quad n \ge n_0$

$50n = O(20n^2 + 14n + 7)$

$50n \le c(20n^2 + 14n + 7)$

$50n \le 20n^2 + 14n + 7 \qquad c = 1$

is true for all $n \ge 2$ $\qquad$ So, $g(n) = O(f(n))$

$c = 1, \ n_0 = 2$

b) By the definition of Big O, $f(n) = O(g(n))$

$0 \le f(n) \le c\,g(n) \quad n \ge 0$

$= f(n) \le c\,g(n) \quad n \ge 0$

$f(n) = O\,g(n)$

$n^2 = O(2^n)$

$n^2 <= c \times 2^n$

$n^2 \le 2^n \qquad c = 1$

$n \le 2^n \qquad n \ge 1$

So, $f(n) = O(g(n)) \qquad c = 1, \ n_0 = 1$

Scanned with CamScanner