Anyans Rathi

Q.3.1 $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \frac{9n}{5n^3} = \frac{9}{5n^2} = 0$

$\lim_{n\to\infty} \frac{g(n)}{f(n)} = \frac{5n^3}{9n} = \frac{5n^2}{9} = \infty$

so $f \in o(g)$
$f \in O(g)$
$g \in \Omega(f)$
$g \in \omega(f)$

(b) $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{9n^{0.8} + 2n^{0.3} + 14\log n}{\sqrt{n}}$

$= \lim_{n\to\infty} = \frac{9n^{0.8}}{\sqrt{n}} + \frac{2n^{0.3}}{\sqrt{n}} + \frac{14\log n}{\sqrt{n}}$

$= 9n^{0.3} + \frac{2}{n^{0.2}} + \frac{14\log n}{\sqrt{n}}$

$= \infty + 0 + 0 = \infty$

so
$f(n) \in \Omega(g)$
$f(n) \in \omega(g)$
$g \in O(f)$
$g \in o(f)$

(c) $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \frac{n^2}{\frac{\log n}{n\log n}} = \frac{n^3 \log n}{\log n} = n^3 = \infty$

$\lim_{n\to\infty} \frac{g(n)}{f(n)} = \frac{n\log n}{\frac{n^2}{\log n}} = \frac{n\log n}{n^2 \log n} = \frac{1}{n} = 0$

$f \in \Omega(g)$
$f \in \omega(g)$
$g \in O(f)$
$g \in o(f)$

(d) $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \frac{(\log(3n))^3}{9\log n} = \infty$

$\lim_{n\to\infty} \frac{g(n)}{f(n)} = \frac{9\log n}{(\log(3n))^3} \cdot 0$

$f \in \Omega(g)$
$f \in \omega(g)$
~~$f \in \Theta$~~ $g \in O(f)$

$g \in o(f)$

Q 3.2 (b) Considering the program.

· The program searches for the minimum element in the unsorted part of the array and swaps it with the current element and as the loop continues to it's next iteration the element gets included in the sorted part of the array.

Loop Invariant in the outer loop: here at the start of every iteration of the outer loop of the program the sub Array A[0...i-1] consists of the smallest elements of the main array A[0...n-1] which are sorted in order.

Initialization
Initially the sorted part of the array has no element in it. when the i=1 then there is only one element. And one element is always sorted in the array size of one.

Maintenance
During maintenance the left part of the sub-array A[0..i-1] is always maintained to be sorted. So in each iteration an element from the unsorted part of the array A[i+1...n-1] or the element with min gets added to the sorted part of the array as the largest element within the sorted part of the array.

(c) For case A and B the sequence input with has been shown in the program all necessary conditions.

(d) Increasing the value from n to 1000 and found the best, average and worst case for each condition. All the values have been stored in "Input.txt". The a GNU graph has been ploted.

(e) Here from the selection sort algorithm and the curve we can interpret tha the differene in the cases is to it's minimal

So for the n times we have

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

so the graph is quadratic for all the cases. The differences generated in the graph is because of other swaps that needed to take place.