Mahiem Agrawal

Problem 3.1

a) $$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{9n}{5n^3} = 0$$

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

Therefore it belongs to.

$f \in o(g)$

$f \in O(g)$

$g \in \Omega(f)$

$g \in \omega(f)$.

b) $$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{9n^{0.8}+2n^{0.3}+1410gn}{\sqrt{n}} = \infty$$

$$\lim_{n\to\infty} \frac{g(n)}{f(n)} = \lim_{n\to\infty} \frac{\sqrt{n}}{9n^{0.8}+2n^{0.3}+1410gn} = 0.$$

Therefore it belongs to the notations.

$\cancel{g}$ $f \in \Omega(g)$

$f \in \omega(g)$

$g \in O(f)$

$g \in o(f)$.

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

$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \frac{n\log n}{\frac{n^2}{\log n}} = \frac{\log n}{n^3\log n} = 0$

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

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

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

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

Problem 3.2

a) The selection Sort has been implemented inside the file "SelectionSort.cpp".

b) To prove that the selection sort is correct we need to show the loop invoriant posses the 3 conditions.

Inictilization : Here we need to show loop invoriant hold prior to the first iteratation. so initialy the sorted array has no element in it so therefore we can say it is sorted at that time.

Maintoinence: Here we need to show that each iteration of the loop maintains the invorient. Here the left part of the sub array till the n element A[0..n] is always sorted and the next element from [n+1 ..... ~~size~~ max] gets added to the left sub-array so it maintains the loop invoriant

Teriminotion: The loop invoriant should ~~show~~ ~~the~~ provide useful propety to show correctness when the loop terimates. At the end of the terimination. the left sub array will reach the size of the loop and there will be no more elements to be sorted.

c) The random input sequence for case A and B have been shown in the code with necessary conditions being shown.

d) ~~En~~ I ~~h~~ I have increased the value of n from 0 to 10,000 and found out the average case, best case, worst case for each condition. The values have been stored in "Input.txt" and later a GNU plot has been made with the help of them.

e) If we take a look at the code regardless of the best case or worst case the number of comparism is always.

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

This shows the complexity to be $\Theta(n^2)$ and the diffrence being generated in the graph is because of the other swap that need to take place.