Algorithms and Data Structures
Assignment #1

Dushan Terzikj

P1:

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

$$\lim_{n\to\infty} \frac{3n}{n^3} = 0$$

$$\Rightarrow \boxed{f(n) \in o(g(n))}$$

This implies that $\boxed{g(n) \in \omega(f(n))}$

b)
$$f(n) = 7n^{0.7} + 2n^{0.2} + 13 \log n$$
$$g(n) = \sqrt{n}$$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$$

Therefore $\boxed{\begin{array}{l} f(n) \in \omega(g(n)) \\ g(n) \in o(f(n)) \end{array}}$ ~~and~~

c) $f(n) = \frac{n^2}{\log n}$
$g(n) = n \log n$

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

Therefore $f(n) \in \omega(g(n))$

$g(n) \in o(g(n))$

d) $f(n) = (\log(3n))^3$

$g(n) = 9\log(n)$

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = \infty \quad \Rightarrow \quad f(n) \in \omega(g(n))$$
$$g(n) \in o(f(n))$$

a) Attached 'selection-sort.c'.

b) How selection sort works is that it searches for the the minimum element in an unsorted part of the array an puts it to the ~~p~~ last position of the sorted part. Let us denote $A[1 \ldots i]$ to be the sorted part of the array. It includes the ~~minimum~~ smallest i numbers of the array A in sorted order (loop invariant). Then we are looking for the minimum element in $A[i+1 \ldots n]$, where n is the length of the array. When that is finished (say j is the position of the minimum element in $A[j+1 \ldots n]$) elements $A[i+1]$ and $A[j]$ are swapped.

Proof by induction:

When we check the first element we have $i=0$

~~Therefore for~~ we look at minimum element in $A[i+1 \ldots n]$, which is the whole array $(A[1 \ldots n])$. Therefore, it is guaranteed that the smallest element in A is at the start after the first ~~ex~~ iteration. (Base case proved).

When we have and index i $(1 \leq i < n)$. We search for the minimum element in $A[i+1 \ldots n]$ and we already know that there is no element smaller in $A[i+1 \ldots n]$ than any element in $A[1 \ldots i]$ because of the loop invariant mentioned before

$\square$

c) Generated sequences are ~~in file~~ generated with the source file "selection_sort_random.cpp". Random sequences are generated with "rand()" function from the C++ <cstdlib> library. Best cases are generated in a way that the array is already sorted. Worst cases are generated in a way that the array is sorted in a decreasing order. More ~~det~~ comments can be found in the source file. The data and the time which was needed for the execution of selection sort is stored in a text file which is later interpreted and plotted using Matlab.

d) Plot can be found in "plot.pdf" file. Blue is randomized sequences, red is best cases and yellow is worst cases

e) The average running time of selection sort is $\Theta(n^2)$.

From the plot we can see that ~~there~~ the blue line makes a parabola of a quadratic equation. This is true for the other two lines as well.

$$g(n) = n^2$$

$$\Theta(g(n)) = \{ f(n) \mid \exists c_1, c_2, n_0 > 0, \text{ such that } 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n), \forall n \geq 0 \}$$

$$c_1 n^2 \leq f(n) \leq c_2 n^2$$

$$c_1 \leq \frac{f(n)}{n^2} \leq c_2$$

In order to have a tight bound, $f(n)$ highest degree has to be 2. Thus, let:

$$f(n) = an^2 + bn + c$$

$$\Rightarrow \quad c_1 \leq a + \frac{b}{n} + \frac{c}{n^2} \leq c_2$$

As $n \to \infty$ we got:

$$c_1 \leq a \leq c_2 \quad \square$$

~~Extra~~ 4

The same goes for $O(g(n))$ and $\Omega(g(n))$, only:

- $O(g(n)) = \{ f(n) \mid \exists c, n_0 > 0$ and $0 \leq \cancel{c} f(n) \leq c g(n), \forall n \geq n_0 \}$
- $\Omega(g(n)) \{ f(n) \mid \exists c, n_0 > 0$ and $0 \leq c g(n) \leq f(n), \forall n \geq 0$