GC05 ALGORITHMIC ANALYSIS COURSEWORK, 2011

Please hand in to 5th floor reception by 12pm on Thursday 24 March

1) Consider two algorithms A and B which solve the same problem, and have time-complexities (in terms of the number of elementary operations they perform) given respectively by $a(n) = 9n + 6$, $b(n) = 2n^2 + 1$.

   (i) Which algorithm is the best *asymptotically*?
   (ii) Which is the best for small input sizes n, and for what values of n is this the case? (You may assume where necessary that n>0.)

2) Are the following statements true or false? Justify your answers using a careful argument based on the formal mathematical definition of 'O' notation. (You may assume where necessary that n is a positive integer.)

   (i) $n^3 \in O(n^2)$
   (ii) $\log_2(2n) \in O(\log_2(n))$
   (iii) $2^n \in O(4^n)$
   (iv) $(n+1)! \in O(n!)$

3) Consider the following short procedures, written in pseudocode. In each case work out f(n), the exact number of unit-time operations the procedure requires as a function of the input size n, simplifying your final answer using O-notation.

   (i)
```
for i  <− 1 to n do
   for j  <− 2 to (n+i) do
      // a unit cost operation
```

   (ii)
```
for i  <− 1 to n do
   for j  <− 1 to n do
      for k <− 1 to (i+j) do
         // a unit cost operation
```

4) Solve the following recurrence relations, simplifying your final answer using 'O' notation. (You may assume that n is a power of 2 where appropriate.)

   (i) $f(0) = 2$
   $f(n) = 6f(n-1) - 5, \quad n > 0$

   (ii) $f(0) = 2$
   $f(1) = 5$
   $f(n) = 5f(n-1) - 6f(n-2), \quad n > 1$

   (iii) $f(0) = 3$
   $f(1) = 12$
   $f(n) = 6f(n-1) - 9f(n-2), \quad n > 1$

   (iv) $f(1) = 3$
   $f(2) = 9$
   $f(n) = 5f(n/2) - 4f(n/4), \quad n > 2$