Tutorial: Complexity

1. Discuss with the aid of examples, the factors that can have an effect on the performance of an algorithm.

2. The following algorithms examined have revealed what their operation count is. Using this as a starting point what is the order (big-O) for each of the algorithms below:
   a) algorithm A: n^4+2n+1
   b) algorithm B: $3n^2+4n+20$
   algorithm C: 2^n+n^2
   c) algorithm D: $n+n\log n$

   Which is the most and least efficient of the algorithms above when
   i) n=10
   ii) n=10000000

3. The table below gives a description of some sorting algorithms.

Name of sort	**worst case**	**memory**
bubble	$O(n^2)$	$O(1)$
selection	$O(n^2)$	$O(1)$
insertion	$O(n^2)$	$O(1)$
shell	$O(n \log^2 n)$	$O(1)$
binary tree	$O(n \log n)$	$O(n)$
merge	$O(n \log n)$	$O(n)$
heap	$O(n \log n)$	$O(1)$
quick	$O(n^2)$	$O(n \log n)$

   a) for a list of `n` elements that need to be sorted, comment of the efficiency of the algorithms. Which is the best, worst and which would you recommend in that case (give reasons).
   i) n = 10
   ii) n = 10000000

4. A square matrix has the same number of rows and columns and its size is defined by the variable **`n`**. The code fragment below performs the multiplication of two square matrices **`A`** and **`B`** and stores the result in matrix **`C`**.

```
int i,j,k,n;
// process rows
for (i = 0, i < n; i++)
  {
  // process columns
  for (j = 0; j < n; j++)
    {
    c[i][j] = 0.0;
    // process row-column interactions and sum them into array c
    for (k = 0; k < n; k++)
      {
      c[i][j] = c[i][j] + a[i][k] * b[k][j];
      }
    }
  }
```

a) Using the code given, step through the statements (as though you were debugging the code) and compute the matrix **C** given that

$$A = \begin{pmatrix} 3 & 5 & 2 \\ 4 & 6 & 1 \\ 1 & -3 & 1 \end{pmatrix} \quad B = \begin{pmatrix} 2 & 1 & 1 \\ 5 & 6 & -3 \\ 3 & 4 & 1 \end{pmatrix}$$

b) What is the computational count ***in terms of n*** for the code. Do this for the case when
   i) you do **not** include the cost of the loop process
   ii) you do include the cost of the loop process

c) Two CPUs CPU1 and CPU2 are being considered to process the matrix multiplication above. Below are the costs of performing standard computational operations:

operation	CPU1 cost (microseconds)	CPU2 cost (microseconds)
add	1	3
subtract	1	3
multiply	4	1
divide	4	1
assign	2	1
compare	1	2

Comparing both CPUs, what is the approximate time needed to compute the triple nested loop for a matrix of size
(i) 3
(ii) 1000

Which CPU is better suited to matrix multiplication?