# Exercise 1

**Data Structures and Algorithms Worksheet**

1. Define the following terms
   1. Algorithm
   2. Data structures
   3. Complexity Analysis
   4. Big-Oh Notation
2. Give a Big-Oh , in terms of *n*, of the running time of each of the following methods.

} }

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a. |  | int ExA( int n ) { int a; | d. | int ExD( int n ) { int a; |
|  | } | for ( int i = 0; i < n; i += 2 ) a = i; |  | for ( int i = 0; i < n \* n; i++ ) for ( int j = 0; j <= n; j++ )  a = i;  } |
| b. | int ExB( int n ) { int a;  for ( int i = 0; i < n \* n; i++ ) a = i;  } | | e. | int ExE( int n ) { int a;  for ( int i = 0; i < n; i++ )  for ( int j = 0; j <= n - i; j++ ) a = i; |
|  |  | |  | } |
| c. int ExC( int n ) { int a;  for ( int i = 0; i < n; i++ ) for ( int j = 0; j <= i; j++ )  a = i; | | | f. | int ExF( int n ) { int a;  for ( int i = 1; i < n; i \*= 2 ) a = i; |

1. Put the followings in order of their complexity
   1. 6*n* log *n* + 2*n*

b. ( 1 / 3 )*n*

1. *n n*
2. log log *n*

# Exercise 2

1. For each of the following two loops:
2. 1 / *n*

f. 210

1. *n* - *n* 3 + 7*n* 5
2. *n* / log *n*
   1. *n* ! [That's *n* factorial.]
3. 2log *n*
4. 2*n*
5. 3*n* + 100 log *n*
   * Write an expression *f(n)* that defines the number of assignment operations executed by the code
   * State what the big-O complexity of the code is
   * Using definition 3, suggest reasonable values for *c* and *N*

i. for (cnt1=0, i=1; i<=n; i++) for (j=1; j<=n; j++)

cnt1++;

* 1. for (cnt2=0, i=1; i<=n; i++) for (j=1; j<=i; j++)

cnt2++;

1. For each of the following two loops, state what the big-O complexity of the code is: i. for (cnt3=0, i=1; i<n; i\*=2)

for (j=1; j<=n; j++)

cnt3++;

ii. for (cnt4=0, i=1; i<=n; i\*=2) for (j=1; j<=i; j++)

cnt4++;

1. The following conversation was overheard between a programmer and a computer scientist: **Programmer:** *I just wrote a really fast program to solve the WOW problem. I wrote it in assembler, optimized every loop, and used every special instruction available on the Quadrium-7 processor.*

**Computer Scientist:** *How fast is it?*

**Programmer:** *Well, for 10 BogoUnits, it took 10 seconds to find the optimum, and for 11, it took 20 seconds, and for 12, it took 40 seconds, and for 13, it took 80 seconds, and so on.*

**Computer Scientist:** *But that's really slow! I have a O( n3 ) algorithm, which is much faster.* **Programmer:** *Faster? I tried your program and it took 1000 seconds for 10 BogoUnits. How can you claim your method is faster?*

* 1. Estimate how long it would take the Computer Scientist's algorithm to solve the problem for 20 BogoUnits.
  2. Estimate how long it would take the Programmer's program to solve the problem for 20 BogoUnits.
  3. What additional information do you need to decide which solution is in fact better for the actual problem?