Algorithm HW#1

1- Which kind of growth best characterizes each of these functions?

	Constant	**Linear**	**Polynomial**	**Exponential**
$3n$				
$3n^2$				
2^n				
$(3/2)^n$				
1000				
1				
$(3/2)n$				
$3n^3$				

2- Rank these functions according to their growth, from slowest growing (at the left) to fastest growing (at the right).

n^2, 2^n, n, n^3, $(3/2)^n$, 1

3- Match each function with an equivalent function, in terms of their Θ. Only match a function if $f(n)=\Theta(g(n))$.

f(*n*)	g(n)
n+30	n^4
n^2 + 2n – 10	3n-1
n^3* 3n	n^2 + 3n
$\log_2 x$	$\log_2 2x$

4- What is the time complexity of the following:

```
y=0
J = 1
   while(j*j<=n):
        y += 1
        j += 1
```

5- Consider the following algorithm for finding the distance between the two closest elements in an array of numbers.

ALGORITHM $MinDistance(A[0..n-1])$
//Input: Array $A[0..n-1]$ of numbers
//Output: Minimum distance between two of its elements
$dmin \leftarrow \infty$
for $i \leftarrow 0$ **to** $n-1$ **do**
for $j \leftarrow 0$ **to** $n-1$ **do**
if $i \neq j$ **and** $|A[i]-A[j]| < dmin$
$dmin \leftarrow |A[i]-A[j]|$
return $dmin$

Make as many improvements as you can in this algorithmic solution to the problem. If you need to, you may change the algorithm altogether; if not, improve the implementation given.

6- Design an algorithm to find all the common elements in two sorted lists of numbers. For example, for the lists 2, 5, 5, 5 and 2, 2, 3, 5, 5, 7, the output should be 2,5,5. What is the maximum number of comparisons your algorithm makes if the lengths of the two given lists are m and n, respectively?

7-

a. Find gcd(31415, 14142) by applying Euclid's algorithm.
b. Estimate how many times faster it will be to find gcd(31415, 14142) by Euclid's algorithm compared with the algorithm based on checking consecutive integers from min{m, n} down to gcd(m, n).

8-

Implement Jupyter (python code)

Good Luck