1. Consider the following algorithm that computes the product of two $n \times n$ matrices, $A$ and $B$, storing the result in $C$.

   for $i \leftarrow 1$ to $n$ do
   for $k \leftarrow 1$ to $n$ do
   $s \leftarrow 0$
   for $j \leftarrow 1$ to $n$ do
   $s \leftarrow s + A(i,j) \cdot B(j,k)$
   $C(i,k) \leftarrow s$

(a) Show that the running time of this algorithm is $O(n^3)$.
(b) Show that the running time of this algorithm is $\Omega(n^3)$.

2. Consider the following algorithm, which operates on an array $A[1\ldots n]$ of integers.

   for $i \leftarrow 1$ to $n$ do
   $A[i] \leftarrow 0$
   for $i \leftarrow 1$ to $n$ do
   $j \leftarrow i$
   while $j \leq n$ do
   $A[j] \leftarrow A[j] + 1$
   $j \leftarrow j + i$

(a) Show that the running time of this algorithm is $O(n \ln n)$.
(b) Describe in words the value of $A[i]$ at the end of execution.

3. The Sieve of Eratosthenes is an algorithm that works on an array $A[2\ldots n]$ of bits, as follows:

   for $i \leftarrow 2$ to $n$ do
   $A[i] \leftarrow 1$
   for $i \leftarrow 2$ to $\lfloor \sqrt{n} \rfloor$ do
   if $A[i] = 1$ then
   $j \leftarrow 2i$
   while $j \leq n$ do
   $A[j] \leftarrow 0$
   $j \leftarrow j + i$

(a) Show that at the end of execution, for $i = 1, \ldots, n$, we have $A[i] = 1$ if and only if $i$ is prime.
(b) It is a fact that

$$\sum_{p \leq n} \frac{1}{p} = \ln(\ln n) + O(1),$$

where the sum is over all primes $p$ up to $n$. Use this fact to show that the running time of this algorithm is $O(n \ln(\ln n))$. 
4. Sort the following functions in order of their rate of growth. That is, sort them into a list \( f_1, f_2, \ldots, f_n \), so that for \( i = 1, \ldots, n-1 \), we have \( f_i = O(f_{i+1}) \). Also, for each adjacent pair of functions \( f_i, f_{i+1} \) in your sorted list, indicate whether or not \( f_i = o(f_{i+1}) \).

\[ n^2, \quad n^n, \quad \ln n, \quad n \ln n, \quad n \ln(n), \quad 20n^2 + 117n - 6, \]

\[ \ln n, \quad \log_2 n, \quad (\log_2 n)^{100}, \quad \log_2(n^{100}), \quad n, \quad 2^{n^2}, \quad n(n + (\ln(n))^2). \]

5. Prove the following facts:

(a) Let \( f(n) = n^\alpha \) and \( g(n) = n^\beta \), for non-negative constants \( \alpha, \beta \). Then \( f = O(g) \) if and only if \( \alpha \leq \beta \), and \( f = o(g) \) if and only if \( \alpha < \beta \).

(b) \( \log_b n = o(n^\alpha) \) for any positive constants \( b, \alpha \).

(c) \( n^\alpha = o(c^n) \) for any constants \( \alpha \geq 0 \) and \( c > 1 \).

6. Prove the following facts:

(a) If \( f = o(g) \), then \( f = O(g) \).

(b) If \( f = O(g) \), then \( cf + dg = O(g) \), where \( c \) and \( d \) are positive constants.

(c) If \( f = O(f') \) and \( g = O(g') \), then \( fg = O(f'g') \).