Assignment 1 Part 1 (due Sunday, May 24, midnight EST)

Instructions:

- Hand in your assignment using Crowdmark. Detailed instructions are on the course website.
- Give complete legible solutions to all questions.
- Your answers will be marked for clarity as well as correctness.
- For any algorithm you present, you should justify its correctness (if it is not obvious) and analyze the complexity.

1. \([8 \text{ marks}]\) Give a proof from first principles (not using limits) the following statements:

   (a) \([4 \text{ marks}]\) \[ n^{2.7} - 100n^{2.4} + 1000 \in \omega(n^{2.5}) \]

   (b) \([4 \text{ marks}]\) Let \(f(n)\) and \(g(n)\) be positive-valued functions. Then:

   \[ \max\{f(n), g(n)\} = \Theta(f(n) + g(n)) \]

2. \([12 \text{ marks}]\) For each pair of functions \(f(n)\) and \(g(n)\), fill in the correct asymptotic notation among \(\Theta, o,\) and \(\omega\) in the statement \(f(n) \in \cup (g(n))\). Formal proofs are not necessary, but provide brief justifications for all of your answers. (The default base in logarithms is 2.)

   (a) \(f(n) = (8n)^{250} + (3n + 1000)^{500}\) vs. \(g(n) = n^{500} + (n + 1000)^{100}\)

   (b) \(f(n) = n^{1.5}2^n\) vs. \(g(n) = (n)^{100}1.99^n\)

   (c) \(f(n) = (256)^{n/4}\) vs. \(g(n) = (125)^{n/3}\)

   (d) \(f(n) = 2^{\log(n)\log(n)}\) vs. \(g(n) = n^{2012}\)
3. [10 marks] Analyze the following pseudocodes and give a tight $\Theta$ bound on the running time as a function of $n$. Carefully show your work.

(a) [5 marks]

1. for $i = 1$ to $n$ do 
2. $A[i] = \text{true}$ 
3. for $i = 1$ to $n$ do 
4. $j = i$ 
5. while $j \leq n$ do 
6. $A[j] = \text{false}$ 
7. $j = j + i$

(b) [5 marks] The following is a sorting algorithm that sorts an array $A$ of $n$ integers, where each integer $e_i \in A$ is $0 \leq e_i \leq m - 1$. Go through the code and verify that this algorithm indeed sorts $A$ correctly.

1. for $i = 0$ to $m - 1$ do 
2. $\text{counts}[i] = 0$
3. for $i = 0$ to $n - 1$ do 
4. $\text{counts}[A[i]]++$
5. $k = 0$
6. for $i = 0$ to $m - 1$ do 
7. for $j = 0$ to $\text{counts}[i] - 1$ do 
8. $A[k] = i, k = k + 1$

4. [12 marks] Given a string $s = a_1a_2...a_n$ of length $n$, where $a_1a_2...a_n \in \{0, 1\}$, decide whether $s$ is the $k$th power of a sub-string $t$, i.e., $s = t^k$, for some $k > 1$ and string $t$. Here, $t^k$ denotes the string $t$ repeated $k$ times. For example, 0100100, 10101010, and 000000, are all perfect powers (e.g. 01000100 = 0100$^2$) but 01000110 is not.

Give an algorithm that solves this problem in $O(n^{3/2})$ time. Describe your algorithm, provide the pseudocode, and analyze the run-time of your algorithm.

Hint: Observe that if $s = t^k$, and $t$ has length $\ell$, then $n = \ell k$. This implies that $\ell$ and $k$ cannot both be greater than $\sqrt{n}$. 
