1. (25 Pts.) Rank the following function by order of growth; that is, find an arrangement $g_1, g_2, \ldots, g_{25}$ of the functions satisfying $g_1 = \Omega(g_2), g_2 = \Omega(g_3), \ldots, g_{24} = \Omega(g_{25})$. Partition your list into equivalence classes such that $f(n)$ and $g(n)$ are in the same class if and only if $f(n) = \Theta(g(n))$:

<table>
<thead>
<tr>
<th>Function</th>
<th>$O(n)$</th>
<th>$\Omega(n)$</th>
<th>$\Theta(n)$</th>
<th>$\omega(n)$</th>
<th>$\Omega(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(3/2)^n$</td>
<td>$(\sqrt{2})^{\log n}$</td>
<td>$\log^* n$</td>
<td>$n^2$</td>
<td>$(\log n)!$</td>
<td>$n!$</td>
</tr>
<tr>
<td>$n^3$</td>
<td>$\log^2 n$</td>
<td>$\log n!$</td>
<td>$2^n$</td>
<td>$n^{1/\log n}$</td>
<td>$\sqrt{2^{\log n}}$</td>
</tr>
<tr>
<td>$\log \log n$</td>
<td>$n.2^n$</td>
<td>$n^{\log \log n}$</td>
<td>$\ln n$</td>
<td>$2^n$</td>
<td>$\log n$</td>
</tr>
<tr>
<td>$2^{\log n}$</td>
<td>$(\log n)^{\log n}$</td>
<td>$4^{\log n}$</td>
<td>$(n+1)!$</td>
<td>$\sqrt{\log n}$</td>
<td>$100$</td>
</tr>
</tbody>
</table>

2. (10 Pts.) Find the error in the following “proof” that $O(n) = O(n^2)$.

Let $f(n) = n^2$, $g(n) = n$, and $h(n) = g(n) - f(n)$. It is clear that $h(n) \leq g(n) \leq f(n)$ for all $n \geq 0$. Therefore, $f(n) = \max(f(n), h(n))$. Thus,

$$O(n) = O(g(n)) = O(f(n) + h(n)) = O(\max(f(n), h(n))) = O(f(n)) = O(n^2)$$

3. (10 Pts.) True or False, Why?: If $f(n) = O(g(n))$ then $2^f(n) = O(2^g(n))$.

4. (20 Pts.) Let $A[1..n]$ be an array of integers of size $n$, where $A[i] + 1 \leq A[i+1]$ for all $1 \leq i \leq n-1$.

(a) Prove by induction that if $A[j] > j$, then $A[k] > k$ for all $j < k \leq n$, and that if $A[j] < j$, then $A[i] < i$ for all $1 \leq i \leq j$.

(b) Design an efficient algorithm to find an $i$ such that $A[i] = i$, if such an $i$ exists. What is the running time of your algorithm?

5. (10 Pts.) Show that $\log n$ is $O(\sqrt{n})$ directly from the definition of $O$-relation.

6. (25 Pts.) For each of the following recurrences, find the asymptotic solution [that is, you do not need to find $T(n)$ precisely, but do need to find $S(n)$ such that $T(n) = \theta(S(n))$]. Assume that $T(n)$ is constant for sufficiently small $n$. Use the master method whenever applicable (and explain which case applies); if it’s not applicable, prove your solution.

(a) $T(n) = 3T(n/2) + n^2$.
(b) $T(n) = 4T(n/2) + n^2 / \log n$ (Hint: you can use the fact that $\sum_{i=1}^k 1/i = (\log k)$).
(c) $T(n) = 2T(n-1) + 1$.
(d) $T(n) = T(n-1)^2$, with $T(0) = 2$.