1. Order of growth (20 points)

Order the following functions according to their order of growth from the lowest to the highest. If you think that two functions are of the same order (i.e. $f(n) \in \Theta(g(n))$), put them in the same group.

$$\log \log n, \log n, \log(n!), n^{1.1}, n \log(n), \sqrt{n}, n^2, n!, (n + 1)!, 2^n, 3^n, 2^{n+1}.$$ 

$$\log \log n << \log n << \sqrt{n} << n \log n \approx \log(n!) << n^{1.1} << n^2 << 2^n \approx 2^{n+1} << 3^n << n! << (n + 1)!,$$

2. Asymptotic Notation (20 points)

For each pair of functions in the table below, determine whether $f(n) \in O(g(n))$, $f(n) \in \Omega(g(n))$, or $f(n) \in \Theta(g(n))$. Complete the table following the examples in the first three lines.

<table>
<thead>
<tr>
<th></th>
<th>$f(n)$</th>
<th>$O$</th>
<th>$\Omega$</th>
<th>$\Theta$</th>
<th>$g(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>i.</td>
<td>$n^2$</td>
<td>X</td>
<td></td>
<td></td>
<td>$n^3$</td>
</tr>
<tr>
<td>ii.</td>
<td>$n$</td>
<td></td>
<td></td>
<td></td>
<td>$\log n$</td>
</tr>
<tr>
<td>iii.</td>
<td>$n^2 - n$</td>
<td>X</td>
<td>X</td>
<td></td>
<td>$5n^2 + 6n$</td>
</tr>
<tr>
<td>a.</td>
<td>$n^2 + 2n$</td>
<td></td>
<td></td>
<td>X</td>
<td>$9n$</td>
</tr>
<tr>
<td>b.</td>
<td>$n + 2\sqrt{n}$</td>
<td>X</td>
<td>X</td>
<td></td>
<td>$2n$</td>
</tr>
<tr>
<td>c.</td>
<td>$n \log n$</td>
<td></td>
<td></td>
<td>X</td>
<td>$n\sqrt{n}$</td>
</tr>
<tr>
<td>d.</td>
<td>$n + \log n$</td>
<td></td>
<td></td>
<td>X</td>
<td>$\sqrt{10n}$</td>
</tr>
<tr>
<td>e.</td>
<td>$3n^2 + 4n + 1$</td>
<td>X</td>
<td></td>
<td></td>
<td>$n^3$</td>
</tr>
<tr>
<td>f.</td>
<td>$2^n$</td>
<td></td>
<td></td>
<td>X</td>
<td>$3^{n/2}$</td>
</tr>
<tr>
<td>g.</td>
<td>$\log(n^3)$</td>
<td>X</td>
<td></td>
<td></td>
<td>$n\log n$</td>
</tr>
<tr>
<td>h.</td>
<td>$2\log^2 n$</td>
<td>X</td>
<td></td>
<td></td>
<td>$\log(n + 1)$</td>
</tr>
<tr>
<td>i.</td>
<td>$\log(n!)$</td>
<td>X</td>
<td>X</td>
<td></td>
<td>$\log(n^n)$</td>
</tr>
<tr>
<td>j.</td>
<td>$\log_{10}(3n + 1)$</td>
<td>X</td>
<td>X</td>
<td></td>
<td>$\log_{3}(n + 5)$</td>
</tr>
</tbody>
</table>

3. Prove asymptotic notation by definition (30 points)

Using the basic definition of $O$, $\Omega$, and $\Theta$, show that

(a) $10n^2 + 2n + 1 \in O(n^2)$

Proof:

By the definition of $O$, $10n^2 + 2n + 1$ is in $O(n^2)$ if there exist positive constants $c$ and $n_0$ such that $10n^2 + 2n + 1 \leq cn^2$ for all $n \geq n_0$.

$$10n^2 + 2n + 1 \leq 10n^2 + 2n^2 + n^2 \leq 13n^2,$$ for all $n \geq 1$.

By choosing $c = 13$ and $n_0 = 1$, we have shown that $10n^2 + 2n + 1 \leq cn^2$ for all $n \geq n_0$. Thus $10n^2 + 2n + 1 \in O(n^2)$ by definition.
(b) $n^2 - 9n + 5 \in \Omega(n)$

**Proof:**
By the definition of $\Omega$, $n^2 - 9n + 5$ is in $\Omega(n)$ if there exist positive constants $c$ and $n_0$ such that $n^2 - 9n + 5 \geq cn$ for all $n \geq n_0$.

$$n^2 - 9n + 5 \geq n^2 - 9n \geq n(n - 9) \geq n,$$ for all $n \geq 10$.

By choosing $c = 1$ and $n_0 = 10$, we have shown that $n^2 - 9n + 5 \geq cn$ for all $n \geq n_0$. Thus $n^2 - 9n + 5 \in \Omega(n)$ by definition.

(c) $3n + 5\sqrt{n} + 2 \in \Theta(n)$.

**Proof:**
To show that $3n + 5\sqrt{n} + 2 \in \Theta(n)$ we can show separately that $3n + 5\sqrt{n} + 2 \in \Omega(n)$ and that $3n + 5\sqrt{n} + 2 \in O(n)$.

First, $3n + 5\sqrt{n} + 2 \geq 3n$ for all $n \geq 0$. Therefore by definition $3n + 5\sqrt{n} + 2 \in \Omega(n)$. ($c = 3, n_0 = 0$.)

Second, $3n + 5\sqrt{n} + 2 \leq 3n + 5n + n \leq 9n$ for all $n \geq 1$ (as $\sqrt{n} \leq n$ when $n \geq 1$). Therefore by definition $3n + 5\sqrt{n} + 2 \in O(n)$. ($c = 9, n_0 = 1$.)

Since $3n + 5\sqrt{n} + 2$ is in both $O(n)$ and $\Omega(n)$, it is also in $\Theta(n)$.

(d) Let $f(n)$ and $g(n)$ be asymptotically positive functions. Prove or disprove each of the following conjectures. (To prove, use definitions of asymptotic notations. To disprove, find a counter example.)

i. $f(n) \in O(g(n))$ implies that $g(n) \in O(f(n) + g(n))$.

**Proof:**
It is easy to see that $g(n) \leq f(n) + g(n)$ for sufficiently large $n$, as $f(n)$ is asymptotically positive (i.e., $f(n) \geq 0$ for sufficiently large $n$). Therefore by definition $g(n) \in O(f(n) + g(n))$. (Note that this is true regardless of the relative asymptotic order of $f(n)$ and $g(n)$.)

ii. $f(n) \in O(g(n))$ implies that $g(n) \in \Omega(f(n) + g(n))$.

**Proof:**
To show that $g(n) \in \Omega(f(n) + g(n))$ we need to show that $g(n) \geq c(f(n) + g(n))$ for some positive constant $c$ and sufficiently large $n$.

Since $f(n) \in O(g(n))$, we know that $f(n) \leq dg(n)$ for some $d > 0$ and all $n$ greater than or equal to some $n_0$. Therefore, we have

$$f(n) \leq d \cdot g(n)$$
$$f(n) + g(n) \leq (d + 1) \cdot g(n)$$
$$(d + 1) \cdot g(n) \geq f(n) + g(n)$$
$$g(n) \geq \frac{f(n) + g(n)}{d + 1}$$

If we choose $c = \frac{1}{d+1}$, we can have $g(n) \geq c \cdot (f(n) + g(n))$. Thus be definition $g(n) \in \Omega(f(n) + g(n))$.

iii. $f(n) \in O(g(n))$ implies that $2f(n) \in \Omega(2^g(n))$.

This statement is false. To disprove, let $f(n) = n$ and $g(n) = 2n$. While $n$ is in $\Omega(2n)$, $2^n$ is not in $\Omega(2^{2n})$. ($2^{2n} = 4^n \in \omega(2^n).$)

4. Bubblesort (20 points).

Bubblesort is a sorting algorithm that works by repeatedly swapping adjacent elements that are out of order. Consider the version of Bubblesort below which sorts the array $A[1 \ldots n]$ into increasing order by repeatedly bubbling the minimum element of the remaining array to the left.
Algorithm 1 bubblesort(int $A[1...n]$)  

\begin{algorithmic}
\STATE $i = 1;$
\WHILE {$i \leq n$} 
\STATE $j = n;$
\STATE 
\STATE //The inner while loop moves the smallest element in $A[i...n]$ to $A[i]$ 
\WHILE {$j > i + 1$} 
\IF {$A[j] < A[j - 1]$} 
\STATE swap $A[j]$ with $A[j - 1]$ 
\ENDIF 
\STATE $j -= 1;$ 
\ENDWHILE 
\STATE $i += 1;$ 
\ENDWHILE 
\end{algorithmic} 

(a) Consider running the above sort on the array $[4, 5, 3, 1, 2]$. Show the sequence of changes which the algorithm makes to the array.

$i = 1$: $[4, 5, 3, 1, 2]$  
$i = 2$: $[1, 4, 5, 3, 2]$  
$i = 3$: $[1, 2, 4, 5, 3]$  
$i = 4$: $[1, 2, 3, 4, 5]$  
$i = 5$: $[1, 2, 3, 4, 5]$  

(b) Give a loop invariant for the outer while loop that will allow you to prove the correctness of the algorithm.

(Hint: consider what is true about the first $i - 1$ entries of the array each time you touch the top of the while loop)

LI: At the beginning of the $i$-th iteration of the outer while loop, the first $i - 1$ entries of the array contains the smallest $i - 1$ elements of the original array in sorted order.

(c) Use induction to prove your loop invariant is true and then use this to prove the correctness of the algorithm. Specifically complete the following:

i. Base case (Initialization): when $i$ is 1, the LI is trivially true because $i - 1 = 0$.

ii. Inductive step (Maintenance): Assume the LI is true before the $i$-th iteration. Then after the $i$-th iteration (i.e., before the $(i + 1)$-th iteration), the inner while loop would have selected the smallest elements in the remaining $(n - i + 1)$ elements and put it in $i$-th position (Note: $A[j - 1]$ is $A[i]$ with $j = i + 1$ at the end of the inner while loop.) So before the start of the $(i + 1)$-th iteration of the outer for loop, the LI remains true with the first $i$-entries containing the smallest $i$ elements in sorted order.

iii. Termination step: When the algorithm terminates, $i = n + 1$. The LI says that the the first $i - 1 = n$ entries have all $n$ elements in the original array and is in sorted order, which means the whole array is now sorted.

(d) Give the best-case and worst-case runtimes of this sort in asymptotic (i.e., $O$) notation.

Both the worst-case and best-case run time are in $\Theta(n^2)$. 