Tutorial 1 solutions

1. These sixteen functions should be ordered as follows:

$$\begin{array}{lllllllll}
17 & < & \log\log n & < & \sqrt{n} & < & \sqrt{n\log n} & < \\
n\log^{17} n & < & n^{18/17} & < & n^3 - n\sqrt{n} & = & n^3 + \log n & < \\
4913n^{289} & < & 1.01^n & < & 3^n & < & 3^{2n-17} & = \\
9^n & < & n! & < & n^n & < & 1.01^{1.01^n} &
\end{array}$$

2. (a) Here we have n iterations of a loop whose body takes constant time to run—regardless of the conditional contained inside—plus some constant-time setup and teardown. Overall, this is $\Theta(n)$.

   (b) The inner loop takes constant time per iteration. The outer loop executes n times, and on the ith iteration the inner loop executes $n - i$ times. This is a common pattern, so you may be able to tell quickly that the overall running time is $\Theta(n^2)$. But let's work it out using sums. The runtime is:

$$\begin{aligned}
\sum_{i=1}^{n}\sum_{j=i+1}^{n} 1 &= \sum_{i=1}^{n}(n-i) \\
&= (\sum_{i=1}^{n} n) - (\sum_{i=1}^{n} i) \\
&= n^2 - \frac{n(n+1)}{2} \\
&= n^2 - \frac{n^2+n}{2} \\
&= n^2/2 - n/2 \\
&\in \Theta(n^2)
\end{aligned}$$

   It's interesting to note also that in the worst case, where the array is in reverse-sorted order, we increment `inversions` $\Theta(n^2)$ times as well, which means there can be $\Theta(n^2)$ inversions in an array of length n.

   (c) This loop isn't just a counting loop. It divides n by two each time. How many times can we divide n by 2 before it reaches 0? Well... an infinite number. Fortunately, we're actually taking the floor of $n/2$, and so we will reach 0.

   If you don't already see where this is going, you might make a table of values of n and number of iterations to guide you toward it:

Initial value of n	Number of loop iterations
0	0
1	1
2	2
3	2
4	3
5	3
6	3
7	3
8	4

At some point, you'll notice that powers of 2 matter and realize that the right side is roughly the log of the left. Specifically, the number of iterations is $\lfloor \lg n \rfloor + 1$ (but just 0 when $n = 0$).

So, we have a logarithmic number of iterations; each iteration takes constant time; plus constant setup and finish time, for: $\Theta(\lg n)$ runtime.