Homework 1

1. Find an exact formula for the number of swaps in bubble sort:

```
void bubble sort (vector <T> & V)
{
  int n = V.size();
  for (int s = n; s > 1; --s)
    for (int i = 0; i < s-1; ++i)
      if (V[i] > V[i+1])
        swap(V[i], V[i+1]);
}
```

A1: Whenever there is a for loop, we will write a summation for the formula. As we can see, there is actually a nested for loop in this algorithm. $_{s = 2}\Sigma^{n} {}_{i = 0}\Sigma^{s - 2}$ (1). The first for loop starts at s = n, then decrements to s > 1, but it basically can be thought of as starting at s = 2, and ending at term n (if you think backwards, inclusive). The next for loop starts at i = 0 and goes on to i < s - 1. The operation that we are analyzing is a function call, swap(). Within the grand scheme of the function call of swap(), we know that only 1 swap takes place, so that's why we have (1) in the summation as the body. We also know that $_{i = first}\Sigma^{last}$ (1) = last - first + 1. So, we use that to solve for the formula. So, the formula is $_{s = 2}\Sigma^{n} {}_{i = 0}\Sigma^{s - 2}$ (1) = $_{s = 2}\Sigma^{n}$ ((s - 2) - 0 + 1) = $_{s = 2}\Sigma^{n}$ (s - 1) = $_{s = 1}\Sigma^{n - 1}$ ((s - 1) + 1) = $_{s = 1}\Sigma^{n - 1}$ (s) = ((n - 1) * n) / 2.

2. Show that

   $2^{\log_3 n} = n^{\log_3 2}$.

A2: $\rightarrow 2^{(\log n / \log 3)} = n^{(\log 2 / \log 3)}$

$\rightarrow \log (2^{(\log n / \log 3)}) = \log (n^{(\log 2 / \log 3)})$

$\rightarrow (\log 2 / \log 3) * \log n = (\log n / \log 3) * \log 2$

$\rightarrow (\log 2 / \log 3) * \log n = (\log 2 / \log 3) * \log n$

3. Which functions grow faster ? Justify your answer.
   - n^{2020} vs. $(3 / 2)^{n}$
   - $n^{1 / 6}$ vs. $(\lg n)^{6}$
   - $n + (\lg n)^{6}$ vs. $n + n^{1 / 6}$

A3:

a) $\lim_{n \to \infty} n^{2020} / (3 / 2)^{n}$. → $\lim_{n \to \infty} (2020 * n^{2019}) / (\ln (3 / 2) * (3 / 2)^{n})$ → $\lim_{n \to \infty} (n!) / ((\ln (3 / 2))^{n} * (3 / 2)^{n}) = \infty$. We learned that it is a fact that factorials will grow faster than any exponential. So, the numerator will increase faster than the denominator. Therefore, the limit, when n approaches infinity, is infinity. This means that $n^{2020} \in \omega((3 / 2)^{n})$.

b) $\lim_{n \to \infty} n^{1/6} / (\lg n)^{6}$. → $\lim_{n \to \infty} ((1 / 6)n^{-5/6}) / (6(\lg n)^{5} * (1 / n (\ln 10)))$ → $\lim_{n \to \infty} ((1 / 6)n^{-5/6}) / (6(\lg n)^{5} * (1 / n (\ln 10))) = 0$. Eventually, $(\lg n)^{6}$ will beat $n^{1/6}$. The denominator will increase faster than the numerator. So, the limit will end up being 0, as n approaches infinity. This means that $n^{1/6} \in o((\lg n)^{6})$.

c) $\lim_{n \to \infty} (n + (\lg n)^{6}) / (n + n^{1/6})$. → $\lim_{n \to \infty} (1 + 6(\lg n)^{5} * (1 / n (\ln 10)) / (1 + (1 / 6)n^{-5/6}) = \infty$. This is the same as b), for when you do l' hospital's rule one more time, the "1 +" on both the numerator and denominator will turn into "0 +" after taking the derivative, negating the effect of the extra "n +" at the beginning of the problem. However, the fraction is flipped, and that means for this problem, the numerator grows faster than the denominator, as n approaches infinity. Then the limit is infinity. This means that $n + \lg n^{6} \in \omega(n + n^{1/6})$.