Do NP-completeness homework, Part 2.

Problem 1.

(a) Is $3^{n+1} = O(3^n)$? Give a brief explaination.

(b) Is $3^{2n} = O(3^n)$? Give a brief explaination.

(c) Show, using limits, that $f(n) = 2n^4 + 3\sqrt{n}$ is $\Theta(n^4)$.

(d) (i) Show, using limits, that $f(n) = n^2$ is $O(2^n)$.
 (ii) Show, using limits, that $f(n) = n^2$ is $\Omega(n)$.

(e) (i) Show, using limits, that $f(n) = n(\log_2 n)^2$ is $O(n^2)$.
 (ii) Find a polynomial function $g(n)$ such that $f(n) = \Omega(g(n))$. Justify.

Problem 2. You wake up in the morning with a great new idea for insertion sort: Instead of inserting an element by stepping down the list one element at a time, you can skip down k elements at a time until you find a smaller element, and then step back up until you find a larger element. So $k = 1$, would be standard insertion sort, and $k = 2$, would mean that you skip every other element. You also realize that for larger i you want to use larger k.

For this problem, we would like to calculate the value of k that minimizes the number of comparisons, when inserting the ith element. Then using this value, figure out how many comparisons this modified insertion sort uses. The exact high order term will suffice. Assume worst case analysis for the entire problem.

It is possible that at some point during your calculations you will need do a summation that you do not recognize. A useful approximation (that often holds) is

$$\sum_{i=a}^{b} f(i) \approx \int_a^b f(x)dx$$

You can assume this. We will justify it Monday.

(a) When inserting the ith element, what value of k should you use? We do not need this exactly, so you can be off by ± 1 (or even 2). Show your work.

(b) Now that you know what value of k to use, give the pseudo-code for your algorithm. Use your judgment on whether to use a sentinel.

(c) How many comparisons does your algorithm use? Just get the high order term exactly. Show your work.

(d) How does this compare with standard insertion sort?