Practice Questions for Week 7

1) An image processing algorithm takes $O(n^3)$ time to run to filter an n x n pixel picture. If it takes 8 seconds to process a 1024 x 1024 pixel picture, how long will it take to process a 1536 x 1536 pixel picture?

2) An algorithm to process an array of size n takes $O(n^2)$ time. If the algorithm takes 113 ms to process an array of size 10,000 how long will it take to process an array of size 100,000, in seconds?

3) A search algorithm on an array of size n runs in O(lg n) time. If 200,000 searches on an array of size 2^{18} takes 20 ms, how long will 540,000 searches take on an array of size 2^{20} take, in milliseconds?

4) An algorithm to process a two dimensional array of size n x m takes O(nmlgn) time. If the algorithm takes 1 second to process an array of size $n = 2^{20}$ by $m = 2^5$, how long will it take to process an array of size $n = 2^{25}$ by $m = 2^9$. Please express your answer in minutes and seconds, with the number of seconds in between 0 and 59, inclusive.

5) An algorithm processing an array of size n runs in $O(n\sqrt{n})$ time. For an array of size 10,000 the algorithm processes the array in 16 ms. How long would it be expected for the algorithm to take when processing an array of size 160,000? Please express your answer in ***seconds,*** writing out exactly three digits past the decimal.

6) Determine the following sum in terms of n: $\sum_{i=1}^{2n-1}(3i-2)$.

7) Let *a*, *b*, *c*, and *d*, be positive integer constants with $a < b$. ***Without using the arithmetic sum formula***, prove that

$$\sum_{i=a}^{b}(ci+d) = \frac{(c(a+b)+2d)(b-a+1)}{2}$$

8) Determine the following summation in terms of n:

$$\sum_{i=0}^{n}(\sum_{j=0}^{i-1} 2^j)$$

9) Determine the following summation in terms of n (assume n is a positive integer 2 or greater), expressing your answer in the form $an^3 + bn^2 + cn$, where a, b and c are rational numbers. (Hint: Try rewriting the summation into an equivalent form that generates less algebra when solving.)

$$\sum_{i=n^2-3}^{n^2+n-4}(i+4)$$

10) Use the iteration technique to determine a Big-Oh solution for the following recurrence relation:

$$T(n) = 4T\left(\frac{n}{2}\right) + n^2, T(1) = 1$$

11) Solve the following recurrence relation defined for non-negative integers, n, using the iteration technique. Please solve the recurrence **exactly**, obtaining a closed-form solution for T(n), in terms of n.

$$T(n) = 2T(n-1) + 2^n, \text{ for n} > 0$$
$$T(0) = 1$$