**2)** (10 pts) ANL (Algorithm Analysis)

(a) (5 pts) 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?

Let the algorithm with input size n take T(n) time where $T(n) = cn^2$. Using the given information, we have:

$$T(10000) = c10000^2 = 113ms$$
$$c = (113 \times 10^{-8} ms)$$

Now, let's plug in n = 100,000:

$$T(100000) = (113 \times 10^{-8} ms) \times 100000^2$$
$$= 113 \times 10^2 ms$$
$$= 11300ms$$
$$= 11.3\ seconds$$

**Grading: 2 pts for solving for c, 2 pts for solving for T(100,000), 1 pt for converting to seconds.**

11.3 seconds

(b) (5 pts) 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?

Let T(n) represent the time for a single search on an array of size n, where $T(n) = clog_2 n$. Using the given information, we have:

$$200000 \times T(2^{18}) = 200000 \times c \times log_2 2^{18} = 20ms$$
$$18 \times 200000c = 20ms$$
$$c = \frac{1}{180000} ms$$

Now, we would like to find 540000T($2^{20}$):

$$540000 \times T(2^{20}) = 540000 \times \frac{1}{180000} ms \times log_2 2^{20}$$
$$= 3 \times 20ms$$
$$= 60ms$$

**Grading: 2 pts for solving for c, 2 pts for solving for result, 1 pt for simplifying**

60 milliseconds