

## 2) (5 pts) ANL (Algorithm Analysis)

An algorithm to process an two dimensional array of size  $n \times m$  takes  $O(nm \lg n)$  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.

Let the algorithm with input array size  $n \times m$  have runtime of  $T(n, m) = cnm \lg n$ , for some constant  $c$ . Using the given information we have:

$$\begin{aligned} T(2^{20}, 2^5) &= c(2^{20})(2^5) \lg 2^{20} = 1 \text{ sec} \\ c(2^{25})(20) &= 1 \text{ sec} \\ c &= \frac{1}{20 \times 2^{25}} \text{ sec} \end{aligned}$$

Now, let's solve for  $T(2^{25}, 2^9)$

$$\begin{aligned} T(2^{25}, 2^9) &= c(2^{25})(2^9) \lg 2^{25} \\ T(2^{25}, 2^9) &= \frac{1}{20 \times 2^{25}} (2^{25})(2^9) \lg 2^{25} \text{ sec} \\ T(2^{25}, 2^9) &= \frac{2^9}{20} (25) \text{ sec} \\ T(2^{25}, 2^{10}) &= \frac{2^9}{4} (5) \text{ sec} \\ \underline{T(2^{25}, 2^{10})} &= 2^7 \times (5) \text{ sec} = 640 \text{ sec} = \underline{\underline{10 \text{ minutes}, 40 \text{ seconds}}} \end{aligned}$$

**Grading: 2 pts solving for  $c$ , 1 pt plugging into solve for new dimensions, 1 pt for some simplification, 1 pt for the final answer in minutes and seconds**