

## CSCI2100: Regular Exercise Set 3

Prepared by Yufei Tao

**Problem 1.** Prove  $\log_2(n!) = \Theta(n \log n)$ .

**Problem 2.** Let  $f(n)$  be a function of positive integer  $n$ . We know:

$$\begin{aligned} f(1) &= 1 \\ f(n) &\leq 2 + f(\lceil n/10 \rceil). \end{aligned}$$

Prove  $f(n) = O(\log n)$ . Recall that  $\lceil x \rceil$  is the ceiling operator that returns the smallest integer at least  $x$ .

**Problem 3.** Let  $f(n)$  be a function of positive integer  $n$ . We know:

$$\begin{aligned} f(1) &= 1 \\ f(n) &\leq 2n + 4f(\lceil n/4 \rceil). \end{aligned}$$

Prove  $f(n) = O(n \log n)$ .

**Problem 4 (Bubble Sort).** Let us re-visit the sorting problem. Recall that, in this problem, we are given an array  $A$  of  $n$  integers, and need to re-arrange them in ascending order. Consider the following *bubble sort* algorithm:

1. If  $n = 1$ , nothing to sort; return.
2. Otherwise, do the following in ascending order of  $i \in [1, n - 1]$ : if  $A[i] > A[i + 1]$ , swap the integers in  $A[i]$  and  $A[i + 1]$ .
3. Recurse in the part of the array from  $A[1]$  to  $A[n - 1]$ .

Prove that the algorithm terminates in  $O(n^2)$  time.

As an example, suppose that  $A$  contains the sequence of integers (10, 15, 8, 29, 13). After Step 2 has been executed once, array  $A$  becomes (10, 8, 15, 13, 29).

**Problem 5\* (Modified Merge Sort).** Let us consider a variant of the merge sort algorithm for sorting an array  $A$  of  $n$  elements (we will use the notation  $A[i..j]$  to represent the part of the array from  $A[i]$  to  $A[j]$ ):

- If  $n = 1$  then return immediately.
- Otherwise set  $k = \lceil n/3 \rceil$ .
- Recursively sort  $A[1..k]$  and  $A[k + 1..n]$ , respectively.
- Merge  $A[1..k]$  and  $A[k + 1..n]$  into one sorted array.

Prove that this algorithm runs in  $O(n \log n)$  time.