Summary 2013 | **CMSC 351: Homework 4** | Clyde Kruskal

Due at the start of class Wednesday, June 12, 2013.

Problem 1. Consider the following recurrence, defined for n a power of 5:

$$T(n) = \begin{cases} 19 & \text{if } n = 1 \\ 3T(n/5) + n - 4 & \text{otherwise} \end{cases}$$

(a) Solve the recurrence exactly using the iteration method. Simplify as much as possible.

(b) Use mathematical induction to verify your solution.

Problem 2. Use the formulas derived in class to obtain exact solutions to the following two recurrences.

(a) Let n be a power of 2.

$$T(n) = \begin{cases} 4 & \text{if } n = 1 \\ 5T(n/2) + 3n^2 & \text{otherwise} \end{cases}$$

(b) Let n be a power of 4.

$$T(n) = \begin{cases} 3 & \text{if } n = 1 \\ 2T(n/4) + 4n + 1 & \text{otherwise} \end{cases}$$

Problem 3. Consider an array of size eight with the numbers in the following order $20, 40, 60, 80, 10, 30, 50, 70$.

(a) What is the array after heap formation? How many comparisons does the standard algorithm use?

(b) Show the array after each element sifts down *after heap creation*. How many comparisons does the standard algorithm use for all of the sifts?

(c) How many comparisons does the modified algorithm (Floyd's version) use to create the heap?

(d) How many comparisons does the modified algorithm (Floyd's version) use for the remainder of the sort?

Problem 4. Give an $O(n \lg k)$-time algorithm to merge k sorted lists into one sorted list, where n is the total number of elements in all the input lists. (*Hint:* Use a min-heap for k-way merging.)

“Master Theorem”

$$T(n) = \begin{cases} aT(n/b) + cn^d & n > 1 \\ f & n = 1 \end{cases}$$

implies

$$T(n) = \begin{cases} \left(f + \frac{c}{ab^{-d}-1}\right) n^{\log_b a} - \frac{cn^d}{ab^{-d}-1} = \begin{cases} \Theta(n^{\log_b a}) & a > b^d \\ \Theta(n^d) & a < b^d \end{cases} \\ n^d(f + c\log_b n) = \Theta(n^d \log_b n) & a = b^d \end{cases}.$$

`Summing solutions:` If

$$T(n) = \begin{cases} aT(n/b) + \sum c_i n^{d_i} & n > 1 \\ f & n = 1 \end{cases}$$

then we can just sum the solutions of each recurrence:

$$T_i(n) = \begin{cases} aT_i(n/b) + c_i n^{d_i} & n > 1 \\ 0 & n = 1 \end{cases}$$

and add in $fn^{\log_b a}$ for the contribution from the leaves.