EECS 340 Algorithms

2018 Spring Semester

Homework 2

Due on February 14, 2018, 12:45pm

This assignment covers recursive thinking.

1. Give a recursive algorithm to find the average (mean) value of an array of 2^k decimal numbers, where $k \in \mathbb{N}$. A run time analysis is not required. $(\star)$

2. Solve R-12.6 in the text, and explain your answer. $(\star)$

3. *Deterministic BogoSort* is a sorting algorithm which achieves the very exciting $\Theta(n!)$ runtime. Moreover, this algorithm is very simple to express in pseudocode:

   Algorithm 1: `det-bogoSort`(L)

   Data: An unsorted list of n integers L
   Result: A sorted copy of L
   for *every possible permutation* $\sigma \in S_n$ *expressing a reordering of elements in* L **do**
   $L' \leftarrow$ A copy of L reordered according to σ
   if L' *is sorted* **then**
   return L'
   end
   end

   For example, to sort $[3, 2, 1]$, this algorithm would check the permuted lists $L' \in \{[3, 1, 2], [2, 3, 1], [2, 1, 3], [1, 3, 2], [1, 2, 3]\}$ in some arbitrary (implementation-specific) ordering, and only **return** after stumbling upon the assignment $L' \leftarrow [1, 2, 3]$. Implement `det-bogoSort` in pseudocode using recursion. $(\star\star)$

4. After leaving CWRU, you find yourself working at Moolah Moneybag's Trading Firm, which sells the BrownianMotion stock trading software to its customers. Your development team has received many requests from users of the backend API to support quick calculation of stock-price moving averages within arbitrary time-windows. Luckily, Mx. Stockholm, the developer who has been working at the

firm for the longest, recalls a long-dead development effort by Mx. Silva to implement this feature. Mx. Stockholm explains that Mx. Silva was obsessed with tree data structures, and left behind the following function:

Algorithm 2: `pow-of-two-average`(S, t_{start}, t_{end})

Data: Stock-price timeseries data S (which is a fancy, bloated wrapper around an array of decimal numbers, indexed by a (small) fundamental time unit), a start time t_{start} and an end time t_{end} (in array indices)

Result: The average stock price from t_{start} (inclusive) through t_{end} (exclusive) $(\frac{1}{t_{end}-t_{start}} \sum_{i=t_{start}}^{i=(t_{end}-1)} S[i])$

Implementation:

... ?

... what?

It's a bunch of gibberish, but you see some references to "Scheme" [a], "interpreter", and "domain-specific language", and for some inexplicable reason there's a *string* in there that's just filled with opening and closing parentheses.

[a] The author of this problem is not hating on Scheme here. If anything, this is a reflective piece on how functional programming evangelism can be pushed too far.

Mx. Stockholm explains that Mx. Silva no longer works for the firm, but this implementation is somehow based on a binary tree of averages being stored with every timeseries. Whatever the case, `pow-of-two-average` only seems to work when $t_{end} - t_{start}$ is a power of two which evenly divides t_{start}. Nevertheless, a code comment says that it has an $O(log(n))$ runtime in terms of the length of the timeseries. Use the existing implementation of `pow-of-two-average` to write pseudocode for a new recursive function `moving-average` which works for any t_{start} and t_{end}. Your method's runtime should be better than linear in the length of the timeseries. $(\star\ \star\ \star)$ [1]

(**Note:** You are permitted to use helper functions, and are also permitted to use statements in English like "the (smallest/largest) multiple of _ (larger/smaller) than _" in place of silly expressions like $k * \lceil \frac{x}{k} \rceil$)

5. Given a matrix $A \in \mathbb{R}^{n \times n}$, we'll call the *checkerboard sum matrix* of A the matrix whose ith column and jth row is given by:

$$\texttt{checker-sum}(A)_{i,j} = (-1)^{i+j} \sum_{k=1}^{i} \sum_{l=1}^{j} (-1)^{k+l} A_{i,j}$$

Write pseudocode for a recursive function to compute $\texttt{checker-sum}(A)_{i,j}$ given the triple (A, i, j). You are not permitted to use any loops (of any kind), nor may you use any form of recursion which emulates one. $(\star\ \star\ \star)$

(**Hint:** Drawing a few pictures may help.)

[1] Bonus Problem: Predict Moolah Moneybag's Trading Firm's short-term $(\star\ \star\ \star\ \star\ \star\ \star\ \star)$ and long-term $(\star)$ stock price trends.

Hint: What is the sound of one hand clapping?

6. **The following is for EECS 454 students only**. Read section 17.1 in the textbook and then solve the following problems:

   (a) R-17.7
   (For the definition of **CLIQUE**, refer to the text preceeding Theorem 17.10)

   (b) C-17.3

Submission

Hand in your paper in-class by the beginning of class on the due date. Always check Canvas for updates and corrections.