CSOR W4231 (sec. 001, 002, H02): Analysis of Algorithms Sep 5, 2019

Homework 1: Due on Sep 19 by 12:01am

Instructors: *Alex Andoni, Cliff Stein*

Instructions. Please follow the homework policy and submission instructions in the Course Information handout. A few highlights to remember:

- follow the collaboration and academic honesty policies;
- write your name, uni, and collaborators on the top page;
- submission is via Courseworks (you are encouraged to use LaTeX, using the provided template);
- if you don't know how to solve a particular part of a problem, just write "*empty*", for which you will get 20% of the points of that part (in contrast, note that non-sensical text may get 0%).

Note that, for each bullet item of a problem, you can use the previous bullets as "black box", even if you didn't manage to solve them. Similarly, you can use anything from the lectures as black-box.

1 Problem 1 (20pts)

For the following pairs of functions f, g, write *all* equations that hold true of the form $f = ?(g)$, where $?$ is one of the $\{o, O, \Theta, \Omega, \omega\}$. Justify your answers (i.e., proofs are required for both why some relations apply and the others don't).

(a) $f(n) = n^4 + 3n^2 - 4n + 1$ and $g(n) = (5n^4 + 1) \cdot (1 + 1/n)$.

(b) $f(n) = 4^{\log_2 n + 0.5}$ and $g(n) = n^2$.

(c) $f(n) = \sum_{i=1}^{n} 1/i^2$ and $g(n) = 100^{1/n}$.

(d) $f(n) = n^{n\%2}$ and $g(n) = n$, where $n\%c$, for an integer c, is n modulo c (e.g., $7\%2 = 1$).

2 Problem 2 (20pts)

Order the following functions by order of growth; that is, if a function f is listed before a function g then $f(n) = O(g(n))$. Group together functions that have the same order of growth, i.e. $f(n) = \Theta(g(n))$. All logarithms are to the base 2, unless explicitly specified otherwise. Justify only the functions that you have grouped together (if any). You do not need to justify the rest of your ordering.

$$\log(n!), \quad \log(n^{0.19}), \quad \binom{n}{n-19}, \quad n^{\log\log n}, \quad 3^{\sqrt{n}}, \quad 2^{(2^{(19+\log n)})}, \quad 2^{(2^{\log n}+19)}, \quad \log(2^{\sqrt{n}})$$

$$2^n, \quad \log n + \log\log n, \quad \pi^{2019}, \quad \log\log n, \quad (\log n)^{\log n}, \quad 2^{\sqrt{\log n}}, \quad n\log n, \quad n^{3+\sin n}.$$

3 Problem 3 (20 pts)

(a) We are given sorted arrays A and B with n integers each, as well as an integer c. The problem is to determine if there exist indexes i, j with $1 \leq i, j \leq n$ such that $2A[i] + 3B[j] = c$ (the output is "yes" or "no"). Design and analyze an algorithm for this problem that runs in linear time (i.e. $O(n)$ time) in the worst case.

(b) Now we have one array A of length n. The problem is to find three integers $i, j, k \in \{1, 2, \ldots n\}$ such that $2A[i] + 3A[j] = A[k]$ (the output is i, j, k or "no"). Design and analyze an algorithm for this problem that runs in $O(n^2)$ time.

4 Problem 4 (20 pts)

Give assymptotic upper bounds on the following recurrences. If the master method applies, you can use that. If not, draw a recursion tree and come up with an upper bound.

(a) $T(n) = 6T(n/9) + n$

(b) $T(n) = 3T(n/3) + n^{1.5}$

(c) $T(n) = T(n-2) + 1/n$

(d) $T(n) = 4T(n/2) + n^2 \lg n$

(e) $T(n) = \sqrt{n}T(\sqrt{n}) + n$

5 Problem 5 (20 pts)

You are given as input an array $A[1..n]$, with n entries. Also, you know that, the array consists of a series of 0's followed by a series of 1's, but you don't know how many of each there are. Your goal is to find the first 1 in the array. There is a cost associated with checking the value of an entry in the array. Each time you check a value and it turns out to be 0, you must pay 1 dollar. If value turns out to be 1, you have to pay 1 "Famous Algorithmist Trading Card (FATC)". In each of the parts of this question, you will have some number of dollars and some number of FATCs. You must give an algorithm which will return the index of the first (lowest indexed) 1 which works given your resources. For each part write pseudocode and explain why your algorithm works, given your resources.

(a) You have n dollars and n FATC

(b) You have 1 dollars and n FATC

(c) You have $10 \log n$ dollars and $10 \log n$ FATC

(d) You have 2 dollars and $10n^{.5}$ FATC

(e) You have k dollars and $10kn^{1/k}$ FATC, for some constant k.