CS 170 Fall 2015 | Algorithms — Prasad Raghavendra and Sanjam Garg | HW 1

Due September 4, 5:00pm

Instructions: You are welcome to form small groups (up to 4 people total) to work through the homework, but you **must** write up all solutions by yourself. List your study partners for homework on the first page, or "none" if you had no partners.

Begin each problem on a new page. Clearly label where each problem (and subproblems, if any) begins. The pages of your homework submissions must be in order (all pages of problem 1 followed by all pages of problem 2, etc.).

For questions asking you to give an algorithm, you must respond in what we will refer to as the "four-part format" for algorithms:

1. High-level description
2. Pseudocode
3. Running time analysis
4. Proof of correctness

Read the relevant Piazza post to understand what these mean.

You risk receiving no credit for any homework that does not adhere to these guidelines.

No late homeworks will be accepted. **No exceptions.** Do not ask for extensions. This is not out of a desire to be harsh, but rather out of fairness to all students in this large course. Out of a total of approximately 11 homework assignments, the lowest two scores will be dropped.

This homework is due Friday, September 4, at 5:00pm via Gradescope. Please submit via PDF, not images.

1. (10 pts.) Getting Started

Please read the posts titled **"Syllabus"** and **"How to do Homework"** on Piazza—there are differences from past semesters! If you have any questions, we are happy to clarify via Piazza. Then, answer the following:

(a) You ask a friend who took CS 170 previously for her homework solutions, some of which overlap with this semester's problem sets. You look at her solutions, then later write them down in your own words. Is this permitted?

(b) True or false: Looking up a problem online in search of an algorithm is permitted, as long as you write it in your words and cite the source.

2. (15 pts.) Compare Growth Rates

In each of the following, indicate whether $f = O(g)$, $f = \Omega(g)$, or both (in which case $f = \Theta(g)$). Briefly justify each of your answers.

	$f(n)$	$g(n)$
(a)	$\log_3 n$	$\log_4 n$
(b)	$n\log(n^4)$	$n^2\log(n^3)$
(c)	$n^{1.04}\log n$	$n(\log n)^3$
(d)	$n\log n$	$(\log n)^{\log n}$
(e)	4^n	$n!$

3. (15 pts.) Bit Counter

Consider an n-bit counter that counts from 0 to $2^n - 1$. For example, when $n = 5$, the counter has the following values:

Step	Value	# Bit-Flips
0	00000	–
1	00001	1
2	00010	2
3	00011	1
4	00100	3
⋮	⋮	
30	11110	2
31	11111	1

For example, the last two bits flip when the counter goes from 1 to 2. Using $\Theta(\cdot)$ notation, find the growth of the *total* number of bit flips (the sum of all the numbers in the "# Bit-Flips" column) as a function of n.

4. (10 pts.) Geometric Growth

Prove that the following formula holds.

$$\sum_{i=0}^{k} c^i = \begin{cases} \Theta(c^k) & \text{if } c > 1 \\ \Theta(k) & \text{if } c = 1 \\ \Theta(1) & \text{if } c < 1 \end{cases}$$

The moral: in asymptotics, the sum of a geometric series is simply the first term if the series is strictly decreasing, the last term if the series is strictly increasing, or the number of terms if the series is unchanging. We will see this idea again when we visit the "master theorem" for solving recurrences.

5. (20 pts.) Universal Hashing

Recall that a *universal family of hash functions* from A to B is a set of functions $\mathscr{H} = \{h : A \to B\}$ such that for any two elements $x \neq y \in A$,

$$\Pr_{h \leftarrow \mathscr{H}}[h(x) = h(y)] = \frac{1}{|B|}$$

where h is drawn uniformly at random from $\mathscr{H}$.

Now consider the following hashing scheme: Let $A = \{0,1\}^n$, $B = \{0,1\}^m$, and $\mathscr{H}$ be the set of all $m \times n$ binary matrices. Compute the hash of a vector $x \in A$ using a matrix $h \in \mathscr{H}$ as $h(x) = hx$, using matrix multiplication modulo 2.

Example. If $h = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}$ and $x = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$, then $hx = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.

Prove that $\mathscr{H}$ is a universal hash family.

Hint: There are multiple approaches. One is to consider the special case when $m = 1$ (so h is a $1 \times n$ matrix), then extend that reasoning to general matrices h.

6. (20 pts.) Mario's Workout

Mario wants to practice his high-jump and needs your help. In this stage of the level, he chooses one of n platforms (p_1 to p_n) to start out on, call it p_a. The platforms are arranged left-to-right, with p_1 on the left, and each platform p_i has a height h_i. Then he jumps to any platform to his right, call it p_b. Mario wants to maximize the elevation difference of his hop ($h_b - h_a$).

Design an efficient algorithm to determine the largest elevation difference Mario can achieve, given an array $H[1..n]$ of the heights of the n platforms.

Hint: There is a slick divide-and-conquer algorithm.

Reminder: Follow the algorithms response format: create separate sections for your algorithm's **main idea**, **pseudocode**, **running time analysis**, and **proof of correctness**.