Algorithms

Christos Papadimitriou and Prasad Raghavendra

HW 1

Due January 30, 5:00pm

Instructions: You are welcome to form small groups (up to four people) to work through the homework, but you **must** write up all your solutions strictly by yourself, and you must acknowledge any ideas you got from others (including from books, papers, web pages, etc.) Please read the collaboration policy on the course web page. List your study partners for homework on the first page, or "none" if you had no partners.

Each problem should begin on a new page. Each page should be clearly labeled with the problem number and the page number of the problem (e.g. Problem 5, page 2 out of 3). The pages of your homework submissions must be in order (all pages of problem 1 in order followed by all pages of problem 2 in order, etc...). You risk receiving no credit for any homework that does not adhere to these guidelines.

On homeworks, we insist that you provide a clear explanation of your algorithms. It is not acceptable to provide a long pseudocode listing with no explanation. In our experience, in such cases the algorithm usually turns out to be incorrect. Even if your algorithm turns out to be correct, we reserve the right to deduct points if it is not clearly explained. We will not grade messy or unreadable submissions. If a problem can be interpreted in more than one way, clearly state the assumptions under which you solve the problem. In writing up your homework you are allowed to consult any book, paper, or published material. If you do so, you are required to cite your source(s). Model solutions will be made available after the due date.

No late homeworks will be accepted. No exceptions. Please don't ask for extensions. We don't mean to be harsh, but we prefer to make model solutions available shortly after the due date, which makes it impossible to accept late homeworks. Out of a total of approximately 12 homework assignments, the lowest two scores will be dropped.

This homework is due Friday, January 30, at 5:00pm electronically. You need to submit it via Gradescope. Please ask on piazza for details on Gradescope and format.

1. (5 pts.) Getting started
Please read the course policies on the web page, especially the course policies on collaboration. If you have any questions, contact the instructors via piazza. Once you have done this, please write "I understand the course policies." on your homework to get credit for this problem.

2. (15 pts.) Compare Growth Rates. In each of the following situations, indicate whether $f = O(g)$, or $f = \Omega(g)$, or both (in which case $f = \Theta(g)$). Give a one sentence justification for each of your answers.

	$f(n)$	$g(n)$
(a)	$n^{1.5}$	$n^{1.3}$
(b)	2^{n-1}	2^n
(c)	$n^{1.3\log n}$	$n^{1.5}$
(d)	3^n	$n2^n$
(e)	$(\log n)^{100}$	$n^{0.1}$
(f)	n	$(\log n)^{\log\log n}$
(g)	2^n	$n!$
(h)	$\log(e^n)$	$n\log n$
(i)	$n+\log n$	$n+(\log n)^2$
(j)	$5n+\sqrt{n}$	$\log n+n$

3. (20 pts.) **G**iven an array $A[1...n]$ of integers we would like to find the i and j such that $\sum_{k=i}^{j} A[k]$ is maximized. Note that the integers in the array may be negative or positive. Consider the following divide-and-conquer algorithm to solve it:

maxSub($A[1,...,n]$)

SUM = 0, CL = $-\infty$, CR = $-\infty$

for $k := n/2+1$ **to** n

SUM $+= A[k]$

CR = **max**(CR,SUM)

SUM = 0

for $k := n/2$ **to** 0

SUM $+= A[k]$

CL = **max**(CL,SUM)

return **max**(maxSub ($A[1,...,n/2]$), CR+CL, maxSub ($A[n/2+1,...,n]$))

What is its running time?

4. (20 pts.) Let D be some large set of data items. A hash function with n buckets is a function $h : D \to \{0,1,\ldots,n-1\}$. Recall that a set of hash functions H is called *universal* if for every fixed pair of data items $x, y \in D$ when we choose a hash function $h \in H$ uniformly at random we have

$$\Pr[h(x) = h(y)] = \frac{1}{n}$$

i.e. the probability of a collision between x and y is $\frac{1}{n}$.

Consider the following alternative property for a set of hash functions H: For every fixed data item $x \in D$ and bucket $i \in \{0,\ldots,n-1\}$ when we choose $h \in H$ uniformly at random we have

$$\Pr[h(x) = i] = \frac{1}{n}$$

i.e. the probability of x landing in each bucket is $\frac{1}{n}$. Suppose we wish to hash m data items. Show that with this alternative property, the average number of elements colliding in some bucket can be as large as m.

5. (20 pts.) Prove the following:

(a) $\sum_{k=1}^{n} k^2 = \Theta(n^3)$

(b) $\sum_{k=1}^{n} \frac{1}{k} = \Theta(\log n)$

(c) $\log(n!) = \Theta(n \log n)$

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

(Hint: Each of these is a big sum or product. Try to make a clever substitution of somewhat smaller terms to get a lower bound, and then do the opposite to get an upper bound.)

6. (20 pts.) Consider the following generalization of the Fibonacci sequence:

- Choose k numbers $a_0, \ldots, a_{k-1}$ and set $F_n = a_n$ for $n < k$.
- Set $F_n = \sum_{i=n-k}^{n-1} F_i$ for $n \geq k$.

Design an algorithm to compute F_n using $O(\log n)$ $k \times k$ matrix multiplications.

(Hint: First try generalizing the algorithm from class to the case $k = 3$.)