Algorithms – CS-27200/37000 Homework – January 19, 2004

Instructor: László Babai Ry-164 e-mail: laci@cs.uchicago.edu

ADVICE. Take advantage of the TA's office hours **Monday, Tuesday and Thursday 5–6pm** in the Theory lounge (Ry–162). Note that a **Monday office hour has been added.**

WEB PAGE. A course web page has been set up at
`http://www.classes.cs.uchicago.edu/archive/2004/winter/27200-1/`
You can navigate here from the department's web page by clicking on Courses, then scrolling to this course and clicking on "Course Web Page." Please check this web page for possible updates the day after each class.

First quiz: Wed Jan 21. No text, no notes, no scratch paper. Calculators are permitted but won't be of much use.

READING Review all previous handouts and find the relevant chapters in the text.

HOMEWORK. Please **print your name on each sheet.** Print "U" next to your name if you seek 27000 credit and "G" if you seek 37000 credit. Please try to make your solutions readable. Unless expressly stated otherwise, all solutions are due at the **beginning of the next class.**

Homework is collected in three separate piles (U, G, "G only"). Please write your solutions to graduate problems on **separate sheets.**

6.1 (U,G) We say that a list of real numbers is sorted with $\le k$ errors if one can remove k items such that the remaining list is sorted. For instance, the list 1, 3, 7, 4, 8 is sorted with one error (remove 7, or remove 4); the list 1, 6, 4, 5, 8, 7 is sorted with two errors (remove 6 and 8, or remove 6 and 7).

(a) (7 points) Given an array $A[1 \dots n]$ of reals, sorted with one error, sort the array in $O(n)$ steps (comparisons and bookkeeping (link updates)). Describe your algorithm in English as well as in PSEUDOCODE. Don't forget to DEFINE your variables.

(b) (G only, 7 points) Given an array $A[1 \dots n]$ of reals, sorted with $\le k$ errors, sort the array in $O(n + k \log k)$ steps. Describe your algorithm in English as well as in PSEUDOCODE.

(c) (G only, 7 points) The *depth* of a sorting network is the number of rounds of parallel comparisons made by the network. Professor Mixup constructed a sorting network which he claims correctly sorts any input which is sorted with one error. Professor Mixup's sorter has depth $O(1)$. Prove that Professor Mixup is wrong; in fact, any sorting network that solves this problem must have depth $\Omega(\log n)$.

6.2 (U,G) (8 points) We are given an array of real numbers $x[1], \dots, x[n]$. The sum of the interval $[i, j]$ is the quantity $S[i, j] := \sum_{k=i}^{j} x[k]$. Find the maximum interval sum $S_{\max}$. Find this value in *linear* time (i. e., the number of operations should be $O(n)$). Describe your solution in pseudocode.

(The solution should be very simple, no more than a few lines. **Elegance counts.** *Hint*: dynamic programming.)

Note: you are not required to output the interval with the maximum sum, just the value of the maximum sum. Observe the following convention:

Convention. If $j < i$, we say that the interval $[i, j]$ is *empty*; the sum of the empty interval is zero. Empty intervals are admitted in the problem. Therefore $S_{\max} \geq 0$ even if all the $x[i]$ are negative.

6.3 (G only, 16 points. Due Monday, February 9.) *This is a difficult problem, start working on it right away.*

We have n coins; at least one of them is bad. All the good coins weigh the same, and all the bad coins weigh the same. The bad coins are lighter than the good coins. Find the *number* of bad coins, making $O(\log^2 n)$ comparisons on a balance. ($\log^2 n := (\log n)^2$.)

Hint. Solve the following auxiliary problem: divide the coins as evenly as possible, using $O(\log n)$ comparisons on the balance. (I.e., if n is even and b, the number of bad coins, is also even, then split the coins evenly, i. e., each tray should have $n/2$ coins, including $b/2$ bad ones. If n is even but b is odd, divide the coins into two groups of $n/2$ coins each so that the number of bad coins in one group is $(b-1)/2$ and in the other, $(b+1)/2$. If n is odd, remove a coin and split the rest as above.) (8 points for the auxiliary problem and another 8 points for showing how to use the auxiliary problem to solve the main problem.)

OPEN QUESTIONS (as far as I know). Do $o(\log^2 n)$ comparisons suffice for the main problem? Do $O(\log n)$ comparisons suffice?

6.4 (6 points, G only, due Friday, January 30) Prove: it takes fewer than $3n/2$ comparisons to find both the maximum and the minimum of n keys from a linearly ordered universe.

6.5 ("Zero-one principle for sorting networks," G only, 6 points, due Friday, January 30) Prove: if a network of comparators correctly sorts all $(0, 1)$-inputs (all lists of zeros and ones) then it correctly sorts all inputs. *Hint.* Prove and use the following observation. *Lemma.* Let f be a monotone nondecreasing real function (i. e., $a < b$ implies $f(a) \leq f(b)$). If on input $(a_1, \ldots, a_n)$ a network of comparators outputs $(b_1, \ldots, b_n)$ then on input $(f(a_1), \ldots, f(a_n))$, the same network outputs $(f(b_1), \ldots, f(b_n))$.