Algorithms CMSC-27200/37000 First Midterm Exam.
February 4, 2004
Instructor: László Babai

Show all your work. **Do not use book, notes, or scrap paper.** Write your answers in the blue books provided. When describing an algorithm in pseudocode, **explain the meaning of your variables** (in English). This midterm contributes 16% to your course grade.

0. (a) (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.

   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.

   WARNING: Half the credit goes for a clear and succint **definition of your variables.** Note that expressions like "current interval" have no meaning unless you have introduced a variable called "interval" which is somehow initialized and updated.

   Note: you are not required to return (output) the interval with the maximum sum, just the value of the maximum sum.

   (b) (G only, 4 points) Modify the pseudocode so that you will return the optimal interval.

1. (3+3 points) Let $F(n)$ denote the number of phases (parallel rounds) taken by Batcher's Odd-Even sorting network. (a) State the recurrence satisfied by $F(n)$. (b) Assuming $n = 2^k$, prove that $F(n) \sim (\log n)^2/2$.

2. (G only, 4 points) State and prove the (0,1)-principle for sorting networks.

3. (3+3 points) (a) Describe in pseudocode how to build a heap from a given array of n reals in $O(n)$ time. (b) Prove that your procedure indeed works in $O(n)$.

4. (5 points) Describe in pseudocode the algorithm that computes the number of connected components of an undirected graph in linear time. Include the code for BFS.

5. (3 points) Recall that a digraph is *strongly connected* if every vertex is accessible from every vertex. State a very simple algorithm that decides in linear time whether or not a graph is strongly connected. You may use BFS as a building block but any other algorithm you use, whether discussed in class or not, you need to describe in pseudocode.

6. (2+2+2 points) (a) State the names of the links associated with each node in a binary tree. (b) Give an array implementation of the links for a completely balanced binary tree on n nodes. (c) (G only) Show that the array implementation can be exponentially wasteful if the binary tree is not balanced.

7. (2 points each) Which of the following sequences are polynomially bounded? To those which are, state the "best" exponent (infimum of the valid exponents). Do not prove. For those which are not polynomially bounded, prove this fact. (a) $\sqrt{n!}$; (b) $\binom{n}{5}$; (c1) $\log(n!)$; (c2) $n^5 \log n + n^7 \cdot 1.1^{-\sqrt{n}}$; (d) (G only) $2^{(\log\log n)^2}$; (e) (G only) $(1+1/n)^{n\log n}$.

8. (G only, 4 points) Let $f_1, f_2, \dots$ be functions, $f_k : \mathbb{N} \to \mathbb{R}$. ($\mathbb{N}$ is the set of nonnegative integers.) True or false: if for all k, $\quad f_k(n) = O(n)$ then $\sum_{k=1}^{n} f_k(n) = O(n^2)$. Prove your answer.

9. (3 points) Give a formal definition of the Knapsack problem. Use mathematical expressions with very few English words; those words should be technical terms (e.g. input, real variable, constraint, etc.) (words like weight, value, knapsack, etc should be avoided).

10. (5 points) State the three data structure operations required for an implementation of Dijkstra's algorithm. State the number of times each operation is called in terms of n = the number of vertices and m = the number of edges of the input digraph. Analyze the total cost under (a) the array implementation (b) the heap implementation of the required priority queue datastructure.