Algorithms – CS-27200/37000 Homework – February 27, 2004

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

ADVICE. Take advantage of the TAs' office hours Monday, Tuesday and Thursday 5–6pm in the Theory lounge (Ry–162).

DATES TO REMEMBER. Mon Mar 8: Midterm 2. Fri Mar 12: **Last class. ATTENDANCE REQUIRED.** Review for final exam. Mon Mar 15, 10:30–12:30: Final Exam

READING. Floyd-Warshall algorithm (all pairs shortest path, transitive closure.) (Text, pp. 629-635.)

GRADUATE READING. Depth-first search (DFS). (Classification of the edges by DFS. The "white-path theorem.") Topological sort.

16.1 (6 points) Let G be an undirected graph (given, as usual, by an array of adjacency lists). Design an algorithm which decides whether or not G contains a cycle and if it does, finds one (outputs the sequence of vertices in the order they occur in the cycle). Your algorithm should run in linear time.

16.2 (4 points) Give a simple formula describing a function $f(n)$ of *intermediate growth*: $f(n)$ should grow faster than any polynomial function and slower than any exponential function. Here "exponential function" refers to a function of the form $c^{n^{1/k}}$ for any constants $k, c > 1$. The terms "faster" and "slower" growth refer to the appropriate little-oh relation between functions. State $f(n)$ and prove that it satisfies the required asymptotic relations.

16.3 (4 points) Let x, y be integers; let $a = xy$ and $b = (x-1)(y-1)$. Given a, b, compute x and y in polynomial time. Estimate the time required if a and b have at most n bits.

16.4 The SUBSET-SUM problem is defined as follows: the input is a list of positive integers $a_1, \dots, a_k, b$; the question is to decide whether or not there exists a subset $S \subseteq \{1, \dots, k\}$ such that $\sum_{i \in S} a_i = b$. (Example: for input $31, 41, 28, 17, 19, 77$ the answer is "yes" since 41+17+19=77.)

(a) (5 points) Prove that SUBSET-SUM $\in$ NP. State the (polynomial-time verifiable) witness (of the yes-answers). Indicate why it is verifiable in polynomial time (length of input to verification algorithm, estimated running time of verification).

(b) (Grad only, 7 points) Solve this problem in $O(kb)$ steps (a step is an arithmetic operation or a pointer update). Describe your algorithm in pseudocode. (The output should be "yes" or "no," you don't need to find the subset S.)

16.5 (8 points) Read Problem 16.4. It is a fact that the SUBSET-SUM problem stated above is NP-complete. On the other hand, it can be solved in $O(kb)$ steps according to Problem 16.4(b). These two facts would appear to lead to the unlikely conclusion that P=NP. Find the error(s) in the following reasoning.

(a) SUBSET-SUM can be solved in $O(kb)$ steps. (This is true, no error here.)
(b) Hence we solved SUBSET-SUM in polynomial time.
(c) If an NP-complete problem can be solved in polynomial time then all problems in the class NP can be solved in polynomial time.
(d) SUBSET-SUM is NP-complete. (True again, no error here.)
(e) (b), (c), and (d) imply that all problems in NP can be solved in polynomial time.
(f) The conclusion of the preceding line is equivalent to the statement that $\mathrm{P} = \mathrm{NP}$.
(g) But we know that $\mathrm{P} \neq \mathrm{NP}$.

For each line, indicate whether you agree or disagree. Discuss in detail the lines with which you disagree.

16.6 (5 points) Switch lines 3 and 4 in the pseudocode of the Floyd–Warshall algorithm given at the bottom of p. 630 in the text (i. e., switch the order of the first two **for**-loops). Show that this modified code is incorrect. You need to build a counterexample; make it as small as possible. Show the result given by the modified code and the result Floyd–Warshall would give.

16.7 In this problem, all graphs are *undirected*. Recall that a *clique* in a graph is a subset S of the vertex set such that the members of S are pairwise adjacent. The *size* of this clique is $|S|$. Suppose we have a black box which takes as input a pair (G, k) where G is a graph and k is a positive integer. The black box outputs "yes" if G contains a clique of size $\geq k$; "no" otherwise.

(a) (4 points) Given a graph $G = (V, E)$, determine the *maximum clique-size* in polynomial time using this black box. You are allowed to make $\leq \lceil \log(|V| + 1) \rceil$ queries to the black box. (The computation time includes the time it takes to write down the questions posed to the black box.) Describe your algorithm in English. Clarity is paramount.
(b) (G only, 5 points) Given a graph $G = (V, E)$, *find a clique of maximum size* in polynomial time using this black box. You are allowed to make a polynomial number of queries to the black box. Describe your algorithm in pseudocode.

16.8 (G only, 10 points) Let G be a connected graph with weighted edges. The max-weight of a spanning tree is the weight of its heaviest edge. A min-max spanning tree is a spanning tree of smallest possible max-weight. Find a min-max spanning tree in linear time. Prove your timing.