Algorithms – CS-27200/37000 Homework – January 26, 2005

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

NEW! The class now has a website,

http://www.classes.cs.uchicago.edu/current/27200-1

Please check it before you do the homework.

ADVICE. Take advantage of the TA sessions.

CHANGE IN SCHEDULE: TA sessions are held in Ryerson-255, Tuesday and Thursday 5–6pm, Saturday 11am–noon, and (this is new) **Wednesday after class** 12:30–1:20 or 1:30–2:20 depending on demand. Indicate your interest in the Wednesday session to the instructor immediately after class. (The Wednesday evening sessions are discontinued.)

IMPORTANT. If you have not done so yet, please send e-mail to the instructor with your name, major, year, type of credit sought (letter grade, P/F, etc.), list of proof-oriented math courses previously taken; include whether or not you took CMSC-27100 (Discrete Math). In the subject write 27200 info or 37000 info, as appropriate.

HOMEWORK. Please **print your name on each sheet.** Print "U" next to your name if you seek 27200 credit and "G" if you seek 37000 credit. Undergraduates receive the stated number of points as *bonus points* for "G only" problems. – 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 put your solutions to "G only" problems on that pile, and your solutions to other problems on the "U" or "G" pile according to the credit you seek.

9.1 (U,G) (Due Monday, January 31) Let P and Q be statements and S a set of instructions. Consider the loop "**while** P **do** S." Recall that Q is a **loop-invariant** for this loop if for all configurations X (all possible settings of the variables) it is true that

> if $P \& Q$ holds for the configuration X then Q also holds for the configuration $S(X)$,

where $S(X)$ is the configuration obtained from X by executing S.

Note that the highlighted statement has to hold even for infeasible configurations X (i. e., for settings of the variables that could not occur in the course of the execution of the algorithm). The situation has some similarity with chess puzzles: when showing that a certain configuration leads to checkmate in two moves, you do not investigate whether or not the given configuration could arise in an actual game.

Dijkstra's algorithm consists of iterations of a single "**while**" loop. Consider the following two statements:

$Q_1 : \quad (\forall u, v \in V)($ if u is black and v is not black then $c(u) \le c(v))$.

$Q_2 : \quad (\forall v \in V)(c(v)$ is the minimum cost among all $s \to \ldots \to v$ paths that pass through black vertices only).

(U,G) Prove that Q_1 is a loop-invariant. Prove that $Q_1 \& Q_2$ is a loop-invariant. (Do not hand in. Zero points.)

(G, 7 points) Prove that Q_2 alone is *not* a loop-invariant. *Explanation.* You need to construct a weighted directed graph with nonnegative weights, a source, and an assignments of all the variables (parent pointers, status colors, current cost values) such that Q_2 holds for your configuration, but Q_2 will no longer hold after executing Dijkstra's **while** loop. Your graph should have very few vertices (4 vertices suffice).

9.2 (U,G) (3+3+3 points) (Due Monday, January 31) For each statement, decide whether or not it is a loop-invariant for BFS: (a) "Vertex #2 is black." (b) "Vertex #2 is white." (c) "Vertex #2 cannot change from black to white." Reason your answers!