Algorithms – CS-27200/37000 Homework – January 19, 2005

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

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 approrpiate.

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.

6.1 (U, G) In this problem, "graph" means undirected graph. Two edges are said to be *independent* if they do not share a vertex. A *matching* in a graph is a set of independent edges. (In other words, a matching in G is a spanning subgraph of G in which every vertex has degree ≤ 1.) A *maximum matching* is a matching of maximum size (max number of independent edges). The *greedy algorithm* for finding a maximum matching proceeds as follows:

Greedy_Matching(G)

The variable M maintains a growing list of independent edges.

```
0 Initialize: M := empty list
1 for e ∈ E(G) do
2     if e is independent of all edges in M then
3         add e to M
4     end(if)
5 end(for)
6 return M
```

(a1) (6 points) Prove: this algorithm does not always return a maximum matching. Show that for every k there exists a graph with maximum matching size $2k$ where the algorithm returns a matching of size k only. (a2) (G only, 3 points) Make your graphs connected.

(b) (G only, 6 points) Prove that the algorithm always returns a matching of size at least half of the maximum.

(c) (2 points) Estimate the number of steps taken by the algorithm in terms of the number of vertices (n) and the number of edges (m). Express your answer using the big-oh notation (ignore a constant factor). If we define $n + m$ to be the input size, is this a "polynomial time algorithms," i. e., is the number of steps polynomially bounded as a function of the input size?

Note that the result of the greedy algorithm depends not only on the graph but on the order in which its edges are accessed.

6.2 (U, G) (Due Monday, January 24. 10 points) Given an undirected graph, either find a legal 2-coloring or find an odd cycle. (See the graph theory handout for the defintions.) Do this in linear time. Describe your algorithm in pseudocode. Prove that your algorithm is correct. *Hint.* Modify BFS.