Algorithms – CS-27200/37000 Homework – March 5, 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

19.1 (7 points) Describe in pseudocode a linear-time algorithm which returns a 2-coloring of G if G is 2-colorable and returns an odd cycle otherwise. (*Hint:* BFS.)

19.2 *Greedy coloring algorithm.* Consider the following algorithm. The algorithm assigns colors (positive integers) to each vertex using a greedy strategy. The color of vertex i is $c(i)$.

```
1    for i = 1 to n
2        let t be the smallest positive integer such that
             (∀j ∈ adj[i])( if j ≤ i − 1 then t ≠ c(j))
3        c(i) := t
4    end(for)
5    return c[1 . . . n]
```

(a) (5 points) Implement the high-level instruction of line 2 so that the algorithm will run in linear time.

(b) (G only, 4 points) Demonstrate the dismal performance of the Greedy Coloring algorithm by exhibiting a bipartite graph for every even value of n such that the algorithm uses $n/2$ colors.

(c) (Research problem, no deadline or point value, but you may earn a lot of credit with the instructor) Let us first perform a random permutation of the vertices and then perform the greedy coloring algorithm. How bad can it get now on bipartite graphs? You need to estimate, from above and from below, the "smallest" $f(n)$ such that for all bipartite graphs and for almost all of the $n!$ permutations, the number of colors used is $\leq f(n)$. (I do not know the answer but I don't think it is terribly hard to obtain reasonable estimates.)

(d) (5 points) Modify the code such that the modified algorithm will 6-color every planar graph.

(e) (5 points) Implement the high-level instructions of (d) so that your algorithm still runs in linear time on planar graphs.