15-750 Algorithms (Spring 2002) **Prof. Manuel Blum**

Homework #8 **due: April 19 (by 11:59pm in WeH 4126)**

Rules: as before.

Problem 1: Looking for a hole (10 points)

(this is Problem A from Section 2.1 of "Programming Pearls", vol. 1, by Jon Bentley)

Given a tape containing a list of 1,000,000 20-bit numbers, find a 20 bit number that is not on the list (why is such a number guaranteed to exist?).

(a) given as much RAM as you need and just the 1 tape containing the million words.

(b) given several tapes and only 2 dozen or so words of memory.

(c) *(M. Blum)* given only the 1 tape containing the million words and just 2 dozen or so words of memory.

In all cases, your algorithm should make just $\mathcal{O}(1)$ passes through the tape(s).

Problem 2: Graph orientation (10 points)

(this problem is due to Jim Propp, Electronic J. Combinatorics, 2002)

Give an algorithm for efficiently solving the following problem:

input: a connected d-regular (undirected) graph G with n vertices.

output: a uniformly random orientation of the edges of G such that the resulting graph has no sink.

"Uniformly random" means that every possible acceptable orientation should be exactly equally likely. (As usual, prove that your algorithm does what it is supposed to do, and analyze it, i.e. determine its expected running time).

(a) Do this for the case that $d = \log n$.

(b) Do it for the case that $d = 2$.

(c) *(extra credit)* Do it for the case that $d = 3$.

Be careful: you must prove that every sinkless orientation is exactly equally likely.

Problem 3: Graph coloring (10 points)

(this problem is due to Abie Flaxman)

Graph 3-colorability is an NP-hard problem. Nevertheless…give an efficient algorithm to properly 3-color the vertices of a 3-colorable graph $G(V, E)$ with n vertices, given that the vertices are all of degree $\geq d$, in the case that:

(a) $d = 2n/3$

(b) $d = n/(\log n)$

Recall: a 3-coloring is a map $f : V \to \{1, 2, 3\}$; a 3-coloring is *proper* if for every edge $(u, v) \in E$, $f(u) \neq f(v)$. (As usual, prove that your algorithm does what it is supposed to do, and …)

Problem 4: Size of a maximum matching (10 points)

(this is Problem 7.7 from M&R; you should solve it <u>without</u> using the reference given in the book)

Let $G(U, V, E)$ be a bipartite graph, and let A be the corresponding matrix of indeterminates as defined in Section 7.3 (Edmonds' matrix). Show that the size of a maximum matching in G is exactly equal to the rank of the matrix A.

Problem 5: Randomized Rank (10 points)

Give a randomizing algorithm to compute the rank of the matrix of indeterminates from Problem 4. (As usual, …)