Algorithms Core 15-750 PROBLEM SET # 6 Spring 1999

DUE: 1:00PM Wednesday April 28 1999

1 Analysis of MIS Algorithm version 2

This problem is worth two problems.

Recall, version 2 of the Luby's Maximal Independent set Algorithm:

Let $G = (V, E)$ be simple undirected graph

Procedure: MIS($G = (V, E)$)

0) If $|V| = 0$ return.
1) For each vertex v pick a random number p_v in the range $[0, 1]$. You may assume the values p_v are distinct.
2) Place v in I if $p_v < p_u$ for all $u \in N(v)$.
3) $V' = V - N(I) - I$ and $E' = \{(u, v) \in E \quad | \quad u, v \in V'\}$
4) return $I * MIS(G' = (V', E'))$

The goal of this problem is to show that with high probability MIS will be called at most $O(\log n)$ times. In parts 2. and 3. below you need not use the constants that I have used.

For each call prove the following:

1. $Prob(v \in I) = \frac{1}{d(v)+1}$
2. Pick a definition of a "good" vertex and a "good" edge. I picked v is **good** if

$$\sum_{u \in N(v)} \frac{1}{d(u)+1} \geq 1/3$$

3. If v is good then $Prob(v \in N(I)) \geq 1/9$
   Hint: Show that if u and w are two distinct vertices which do not share an edge then

$$\text{Prob}[u \in I \quad \& \quad w \in I] = \frac{d(u) + d(w) + 2}{(d(u)+1)(d(w)+1)(d(u) + d(w) + 2 - k)}$$

   where k is the number of vertices common to u and$w$.

3. The expect number of edges deleted is at least $|E|/18$

For each of the parts you may use results from Kozen.

2 Computing the minimum number of registers

Let $\mathcal{T}$ an arbitrary computation tree over the domain of integer $\mathcal{Z}$ with the operation at the internal nodes being $\{\oplus\}$ defined as following:

$$\oplus(a,b) = \begin{cases} a+1 & a=b \\ max(a,b) & otherwise \end{cases}$$

Show that any tree over this operation can be evaluated in $O(\log n)$ time using $n/\log n$ processors.

Hint: Consider the following class of function:

$$\mathcal{F} = \{f_{a,b} \mid a \in \mathcal{Z} \cup \{-\infty\}, b \in \mathcal{Z}, a \leq b\}$$

where

$$f_{a,b}(x) = \begin{cases} a & x < a \\ \max(b,x) & \text{otherwise} \end{cases}$$

3 Parallel Selection

Consider the $O(n)$ deterministic sequential algorithm for section, see page 190 of CLR.

Problem 3.1 *Write this algorithm as a parallel algorithm using the parallel constructs from class without changing the work performed. Analysis the asymptotic time for your parallel algorithm?*

Consider the Randomized-selection algorithm, see page 187 of CLR.

Problem 3.2 *Write this algorithm as a parallel algorithm using the parallel constructs from class without changing the work performed. Analysis the asymptotic expected time for your parallel algorithm? Instead of computing the expect time you may use Chernoff Bounds to bound the running time with high probability.*