Notes on Parallel Maximal Independent Sets

1. Types of Randomized Results.
   a. Las Vegas - execution time is a random variable, but the expectation is bounded w.h.p.

      w.h.p. = with high probability

      A probability bound of at least $1 - 1/n^c$ where c is a constant

   b. Monte Carlo - algorithm fails with some probability

      1. Usually can run algorithm many times to decrease probability of failure past hardware failure rate.

      2. Decision problems usually have a one-sided error - can say ‘‘yes’’ incorrectly, but not ‘‘no’’.

   c. Random generation (coin flips - biased or not) is a significant issue.
2. Classic Probabilitistic/Randomized Algorithms
   a. Quicksort/Selection - Las Vegas

   b. Hashing - Las Vegas

   c. Routing - Las Vegas

   d. List Ranking - Las Vegas

   e. Skip Lists - Las Vegas

   f. Primality Testing - Monte Carlo

   g. Verifying Matrix Products - Monte Carlo
3. Independent Set Problem Definitions
   a. Independent Set - For an undirected graph, a subset of V such that no two vertices are adjacent.

   b. Maximal IS (MIS) - An IS which is not a proper subset of another IS

   c. Maximum IS - IS with maximum cardinality (NP-Complete)

d. Lexicographically-first MIS

1. P-Complete

2. Output from following greedy MIS algorithm is the LFMIS:

```
S := V
IS :=∅
for i := 0 to n - 1
    if i ∈ S
        IS := IS ∪ { i }
        for each edge {i, x}
            S := S - { x }
```

3. Runs in θ(n + m) worst-case time

4. Luby's Probabilistic Algorithm

```
S := V
IS :=∅
while S ≠ ∅
    for each vertex i ∈ S { in parallel }
        Mark i with probability 1/(2d(v))    {d(v) = degree of vertex v}
    for each edge {x, y} with both vertices marked {in parallel}
        if d(x) < d(y)
            Unmark x and remove x from S
        else if d(y) < d(x)
            Unmark y and remove y from S
        else if x > y
            Unmark x and remove x from S
        else
            Unmark y and remove y from S
        IS := IS ∪ marked edges
        Remove marked edges from S
```

W.H.P., at least m/72 edges are expected to be removed in each iteration of while-loop.

θ(log m) time using θ(n + m) processors in CRCW (θ($\log^2$ m) time for EREW).

5. Luby's Deterministic Algorithm

   a. The trick (handwaving)

      Probabilistic version uses up to n random number in each stage.

      Can replace with generating just two numbers, x and y, in each stage (still randomized)

      p = prime number in range n .. 2n

      x and y will be random numbers in range 0 .. p - 1.

      To test whether vertex i is (initially) marked, test whether:

      $(x + yi) < p/(2d(i))$ (roughly)

   b. Time and increase in processors (to make deterministic). Note that there are log n + 1 bits in x and y, so there are

      $2^{2\log n + 2} = 4 \bullet 2^{2\log n} = 4n^2$ possibilities for x and y.

      By increasing the processors to $\theta(4n^2(n + m))$ and having each stage select a (x, y) pair that maximizes the number of edges removed, a deterministic algorithm is obtained.