7/3/2013 **Solution to the 2nd term exam in Distributed Algorithms**

1a) The reduction was described in class.

1b) It cannot be used for $(\Delta+1)$-coloring of unoriented trees. As a result of applying the reduction to an unoriented tree we get a graph which is not a tree. So an MIS algorithm for unoriented trees cannot be applied to this resulting graph.

2) The construction was given in class. In short, draw a graph from $G(n,p)$, $p = n^{1/k}$. With high probability, it has $\Omega(n^{1+1/k})$ edges, and $o(n)$ cycles of length $< k$. Eliminate one edge from each cycle to get the desired

graph.

This construction implies that there exists a constant $c > 0$ so that a (ck)-spanner for general graphs requires $\Omega(n^{1+1/k})$ edges.
The reason is that no edge can be removed from G without increasing some distance by a factor of at least $k-1$. So any $(k-1)$-spanner for G requires $\Omega(n^{1+1/k})$ edges.

3) $(2+\varepsilon)a$-coloring can be computed in $O(a \cdot \log n)$ time.
We need to work out the dependence on $\varepsilon > 0$. During the construction of the H-decomposition, after removing each H-set we are left with at most $\frac{2}{2+\varepsilon}$-fraction of all vertices

that were in the graph before this set was extracted.

Hence the number of H-sets in the decomposition is l s.t.

$$\left(\frac{2}{2+\varepsilon}\right)^{l} \cdot n < 1 \qquad \text{(the smallest } l \text{ that satisfies it).}$$

$$l = O\left(\frac{\log n}{\varepsilon}\right).$$

Set $\varepsilon = \frac{1}{a^{1/3}}$. As a result we get a $\cancel{(2a)(1+\varepsilon)}$ $(2+\varepsilon)a = 2a + O(a^{2/3})$-coloring in time $O(a \cdot \frac{1}{\varepsilon} \cdot \log n) = O(a^{4/3} \cdot \log n)$.

4) <u>Lm:</u> The min-bottleneck-spanning tree is the MST.

<u>Pf</u>:

Let T be the MST, and let T' be some another spanning tree for G. Let e be the heaviest edge in T. If e belongs to T' too, then the heaviest edge e' in T' satisfies

$w(e') \geq w(e)$, as required.

If e does not belong to T' then $T' \cup \{e\}$ contains a cycle C. By the red rule, e is not the strictly heaviest edge in C. So C contains another edge $\hat{e}$ (from T') with $w(\hat{e}) \geq w(e)$. Hence the heaviest edge e' in T' satisfies $w(e') \geq w(\hat{e}) \geq w(e)$, as required. QED

Hence any of the algorithms that we studied for computing the MST does the job (and computes the min-bottleneck-spanning-tree).

By GHS alg' it can be done in $O(n \log n)$ time and $O(|E| \cdot \log n)$ communication.