CSC2421S **Problem Set 1** **Winter/Spring 2005**

Due: Thursday, February 3, beginning of lecture

1. Consider the following greedy algorithm (call it EFT) for the $JISP_1$ problem:

   $S := \emptyset; t := 0$
   While $\mathcal{I} \neq \emptyset$
   $\quad j := argmin_k\{f_k | s_k \geq t\}$
   $\quad S := S \cup \{j\}$; remove I_j from $\mathcal{I}$
   $\quad$ remove any I_k such that $\mu_k = \mu_j$ or $s_k < f_j$
   $\quad t := f_j$
   EndWhile

   (a) Show that there is an input instance $\mathcal{I}$ for which $|OPT[\mathcal{I}]| = 2 \cdot |EFT[\mathcal{I}]|$

   (b) Prove that EFT is a 2-approximation algorithm.

   (c) Research question: Find a deterministic polynomial time algorithm which achieves an approximation ratio < 2. Alternatively or additionally, what is the best lower bound that you can derive for priority (or one-pass) algorithms?

2. Consider the unweighted interval selection problem when presented as the problem of computing a maximum size independent set in an interval graph. Namely, the input items are nodes which are represented by their list of adjacent nodes. Consider the following greedy algorithm (call it $MDEG$) for this problem:

   $S := \emptyset$
   While $\mathcal{I} \neq \emptyset$
   $\quad$ Let $v \in \mathcal{I}$ be a node of (current) minimum degree
   $\quad S := S \cup \{v\}$
   $\quad \mathcal{I} := \mathcal{I} - \{v\} - \{v_i | v_i \text{ adjacent to } v\}$
   EndWhile

   (a) Show that $MDEG$ is a 2-approximation algorithm.

   (b) In fact, $MDEG$ is a $\frac{3}{2}$-approximation algorithm. Show that there exists a sequence of input graphs $G_1, \ldots, G_n$ such that $\lim_{i\to\infty} \frac{|OPT[G_i]|}{|MDEG[G_i]|} = \frac{3}{2}$. (We have presented the graph G_1 in class.) To prove the upper bound you can try to charge intervals (nodes) in OPT to intervals in $MDEG$ so that for certain pairs of intervals in $MDEG$, at most 3 intervals in OPT are charged to these pairs.

   (c) (Minor) research question: What is the best lower bound that you can derive for priority (or one-pass) algorithms?

3. More to follow