Design and Analysis of Algorithms	**Exam 2**
CS681 Fall 2007	**Monday, December 3, 2007**

Instructions

Sign out the exam with Kelly Patwell in 5147 Upson. Return the completed exam to Kelly or submit it via CMS within 72 hours after signing it out, but no later than 4pm Friday, December 14. There are five questions, all equal weight. You may consult class notes, class texts, and handouts posted to the CS681 web page, but nothing else. No collaboration is allowed. Partial credit will be awarded where appropriate, so be explicit. All algorithms should be accompanied by a proof of correctness and complexity analysis. **Good luck!**

---

1. Consider the following video-on-demand problem. There are a large number of movies stored on several servers. Each movie may have copies on several different servers (though not necessarily on all servers). The allocation of movies to servers is fixed. Each server has a fixed *capacity*, which is the number of simultaneous streams it is capable of supporting. Given a set of requests for movies, you would like to assign each request to some server containing that movie so that no server is assigned more than its capacity and the maximum number of requests are served. Give a polynomial-time solution for this problem.

2. Given a directed graph $G = (V, E)$, we say $K \subseteq V$ is a *kernel* for G if

   (a) there is no edge between any two vertices of K, and

   (b) for every $v \in V - K$, there exists $u \in K$ such that $(u, v) \in E$.

   Show that it is *NP*-complete to decide whether a given G has a kernel. (*Hint.* Use $3CNFSAT$. Build a graph with a 2-cycle through each pair of complementary literals, along with some other vertices and edges.)

3. The *two-dimensional discrete Fourier transform* maps an $M \times N$ matrix of complex numbers A to another $M \times N$ matrix of complex numbers $F(A)$. It is defined by

$$F(A)_{ij} \stackrel{\text{def}}{=} \sum_{k=0}^{M-1} \sum_{\ell=0}^{N-1} A_{k\ell}\,\omega_1^{ik}\omega_2^{j\ell},$$

   where ω_1 and ω_2 are primitive M-th and N-th roots of unity, respectively. Argue that this transform is invertible and give a very efficient *NC* algorithm to compute it. (*Hint.* Reduce to the one-dimensional case.)

4. Randomized and deterministic NC algorithms for finding a maximal matching in an undirected graph are given on pp. 229 and 271 of the course text. You may use these algorithms without proof in your solution for this problem.

   Consider the following randomized parallel vertex cover algorithm. Start with $C_0 = \varnothing$. At stage $t \geq 1$, compute a maximal matching M_t. For each edge $e \in M_t$, choose one endpoint of e at random with both choices equally likely. Let C_t be the set of selected vertices. Delete C_t and all incident edges. Repeat until there are no more edges. Let $C = \bigcup_t C_t$ and $M = \bigcup_t M_t$.

   (a) Show that C is a vertex cover.

   (b) Show that the expected number of edges in M adjacent to any vertex is at most 2.

   (c) Conclude that the expected size of C is at most twice the size of a minimum vertex cover.

5. A *mixed graph* is a graph in which some of the edges are directed and some are undirected. Prove that if a given mixed graph has no directed cycle, then it is always possible to orient the undirected edges so that the resulting directed graph is acyclic. Show how to find such an orientation if one exists in (a) linear time, (b) NC.