CMSC 651, Analysis of Algorithms, University of Maryland, Fall 2012
Mid-term exam, due by 11:59PM on Oct 25, 2012

Instructions:

- Solve these problems individually, and NOT with your group or anyone else. You are only allowed to use your own notes and the papers/videos available from the class web-page; other resources including books, the Web etc. are not allowed for consultation. *Write the honor pledge "I pledge on my honor that I have not given or received any unauthorized assistance on this examination" at the start of your answers.*
- There are 5 problems. Write/type your name and solutions neatly.
- Partial credit is possible; if you cannot solve a problem fully but can make some meaningful progress, write your ideas clearly.
- Please email your solutions in PDF to Aravind and Catalin by 11:59PM on Oct 25th; alternatively, you can slip your written solutions under Aravind's office door, or give to him personally in class or during office hours on Oct 25th.

1. Let a k-DOUBLE-CLIQUE denote the following type of structure in a given undirected graph G: "there exist $2k-1$ different vertices, denoted by $u, v_1, v_2, \ldots, v_{k-1}, w_1, w_2, \ldots, w_{k-1}$, such that $\{u, v_1, v_2, \ldots, v_{k-1}\}$ is a k-clique, and $\{u, w_1, w_2, \ldots, w_{k-1}\}$ is also a k-clique".

Let DC denote the problem of deciding, given k and G, whether G has some k-DOUBLE-CLIQUE.

(i) Prove that DC is in NP. **(3 points)**

(ii) Prove that DC is NP-complete. **(10 points)**

2. In simplified language, many divide-and-conquer graph algorithms are of the following type: *given a graph $G = (V, E)$ with n vertices, spend $O(n \log n)$ time to find some (suitably good) partition of V into two subsets A and B with $|A| = n/3$ and $|B| = 2n/3$, and then recursively solve the problem separately for A and B.* (As usual, if n is small – if $n \leq 3$, say – then we solve the problem directly on G in constant time.)

(a) Write a recurrence relation for the time complexity of such algorithms. **(3 points)**

(b) Solve this recurrence with proof. **(10 points)**

3. Let us call a vector $y = (y_1, y_2, \ldots, y_n)$ of real numbers "m-almost-constant" if there exist at most $m-1$ values of $i \in \{1, 2, \ldots, n-1\}$ such that $y_i \neq y_{i+1}$. Develop a polynomial-time algorithm which, given a vector $x = (x_1, x_2, \ldots, x_n)$ of real numbers, finds the smallest m for which there exists some m-almost-constant vector $y = (y_1, y_2, \ldots, y_n)$ such that $|x_i - y_i| \leq 5$ for all i. **(10 points)**

4. You are given a set of jobs $V = \{1, 2, \ldots, n\}$ and p processors, in the following context: (i) there is a given *directed acyclic* graph $G = (V, E)$ with the jobs as vertices, such that if $(u, v) \in E$, then job u must be finished before starting job v, and (ii) each job takes one unit of time to process, and can be processed by any machine. For instance, if $n = 4$, $E = \{(1, 3), (2, 3), (3, 4)\}$ and $p = 2$, then a legal schedule is one in which jobs 1 and 2 are processed by the two processors separately in time-step 1, job 3 at time-step 2, and job 4 at time-step 3 (the last two processing steps can be done by any of the two processors).

Let $u \to v$ denote that there is a directed path from u to v in G. Call a set of vertices S a *chain* if the vertices of S can be numbered as $u_1, u_2, \ldots, u_{|S|}$ such that $(u_i, u_{i+1}) \in E$ for all i; call S an *admissible set* if $u \not\to v$ for any pair of distinct vertices u and v in S. Let ℓ denote the maximum cardinality of a chain in G (e.g., we have $\ell = 3$ in the example of the previous paragraph).

(a) Prove that V cannot be partitioned into less than ℓ admissible sets. **(5 points)**

(b) Show how to partition V efficiently into ℓ admissible sets. **(6 points)**

(c) Show that all the jobs can be scheduled (processed) within $\frac{n}{p} + \ell$ time steps. (Hint: when can the processors work simultaneously on a set of jobs?) **(8 points)**

5. Given an undirected graph $G = (V, E)$, a k-*coloring* of G is any function $f : V \to \{1, 2, \ldots, k\}$ such that if $(u, v) \in E$, then $f(u) \neq f(v)$. It is known that checking if a given graph is k-colorable, is NP-complete for $k \geq 3$. Suppose instead that we only want that $f(u) \neq f(v)$ for at least some α-fraction of the edges (u, v), for some positive value $\alpha \in (0, 1]$. (α will be a function of k.) Design an efficient randomized algorithm for this: i.e., construct a function $f : V \to \{1, 2, \ldots, k\}$ in random polynomial time such that, for some $\alpha \in (0, 1]$, the expected number of edges (u, v) for which $f(u) \neq f(v)$, is at least $\alpha|E|$. Analyze your algorithm, and show how large a value of α it guarantees. (Note that the graph G is arbitrary; the only randomness is from the algorithm.) **(5 points)**