READ THE INSTRUCTIONS POSTED IN THE CLASS HOMEPAGE
http://www.cc.gatech.edu/~mihail/index4540.html
BEFORE AND WHILE COMPLETING THIS EXAM.

Problem 1
An array $a_0 \ldots a_{n-1}$ is said to have significant repetition if and only if there exists at least one $x$ such that $x$ appears more than $n/4$ times: $|\{i : a_i = x\}| > n/4$. Given an array, the task is to design an efficient algorithm to tell whether the array has significant repetition, and output all $x$’s such that $|\{i : a_i = x\}| > n/4$. The elements of the array are not necessarily from some ordered domain like the integers, and so there can be no comparisons of the form ”$a_i > a_j$?” (Think of the array elements as GIF files, say.) However you can answer questions of the form ”$a_i = a_j$?” in constant time. Show how to solve this problem in $O(n \log n)$ time.

Problem 2
Let $A = a_1 \ldots a_n$ be a set of distinct elements, where each element is colored from a set of ten distinct colors. Think of $n$ as very large, e.g. the size of an huge population database, the WWW, etc. Think of the set of colors as trends, preferences of the population elements, etc.

• Say that a color $x$ is important in $A$ if $|\{i : a_i = x\}| > n/3$.
• Say that a color $x$ is unimportant in $A$ if $|\{i : a_i = x\}| < n/5$.

Let $\epsilon$ be a small number denoting failure probability, $\epsilon \in (0, 1)$. Consider an algorithm that collects $k = c \ln \left(\frac{2}{\epsilon}\right)$ independent samples from $A$, for some constant $c$ independent of $n$, and outputs:

• A color $x$ marked as significant in $A$, if $x$ occurred more than $k/4$ times in the sample.
• A statement no significant color in $A$, if every color occurred less than $k/4$ times in the sample.

Determine $c$ so that (a) and (b) below hold:

(a) The probability that, some color $x$ is important in $A$, but the algorithm output ”no significant color in $A$”, is at most $\epsilon$.
(b) The probability that all colors are unimportant in $A$, but the algorithm output a color $x$ marked as ”significant in $A$”, is at most $\epsilon$.

Problem 3
Consider the usual dynamic programming algorithm, with running time $O(n^3)$, that computes all pairs shortest paths in graphs with non-negative edge-costs.

(a) Now suppose that $n$ processors are available to work in parallel. Suppose that processors have access to each other’s memories, and suppose that there is one additional higher level processor that ensures synchronization. Argue that in this parallel setup, the all pairs shortest path algorithm can be implemented, so that it runs in $O(n^2)$ steps.

(b) Give an example of another dynamic programming algorithm, and explain how using processors running in parallel can substantially improve its running time.
Problem 4
Vertex Coloring is the following problem: Given an undirected graph $G(V,E)$, color its vertices with the minimum number of colors so that the two endpoints of each edge receive distinct colors.
(a) Give a polynomial time algorithm that decides if a graph can be colored with with two colors so that the two endpoints of each edge receive distinct colors.
(b) Give a polynomial time algorithm for coloring a graph with $\Delta+1$ colors, where $\Delta$ is the maximum degree of a vertex in $G$.

Problem 5
Consider the following integer program (IP), where all the $c_i$'s, $a_{ij}$'s and $b_j$'s are non-negative.

\[
\begin{align*}
\text{(IP)} \\
\min & \sum_{i=1}^{n} c_i x_i \\
\text{such that} & \quad \sum_{i=1}^{n} a_{ij} x_i \geq b_j \quad 1 \leq j \leq m \\
x_i \in \{0, 1\} \quad 1 \leq i \leq n
\end{align*}
\]

Write the (LP)-relaxation of the above (IP). Let $x_i^*, 1 \leq i \leq n$, be an optimal solution to the (LP)-relaxation, such that

$$x_i^* \in \left\{0, \frac{1}{2}, 1 \right\}.$$ 

Such a solution is called \textit{half integral}.

Show how to convert the above half integral solution of the (LP)-relaxation to a solution $y_i, 1 \leq i \leq n$, of (IP) such that

$$\sum_{i=1}^{n} c_i y_i \leq 2\text{OPT(IP)}.$$