Programming Problem 1 (10 Points):
The Chvatal-Sankoff constants are mathematical constants that describe the length of the longest common subsequences of random strings. Given parameters $n$ and $k$, choose two random length $n$ strings $A$ and $B$ from the same $k$-symbol alphabet, with each character chosen uniformly at random. Let $\lambda_{n,k}$ be the random variable whose value is the length of the longest common subsequence of $A$ and $B$. Let $E[\lambda_{n,k}]$ denote the expectation of $\lambda_{n,k}$. The Chvatal-Sankoff constant $\gamma_k$ is defined at

$$\gamma_k = \lim_{n \to \infty} \frac{E[\lambda_{n,k}]}{n}.$$ 

Experimentally determine (by implementing an LCS algorithm), the smallest value of $k$, such that $\gamma_k < \frac{2}{5}$. In other words determine how large an alphabet needs to be so that the expected length of the LCS of two random strings is less than 40% the length of the strings.

a.) Generate a table of estimates for $\gamma_k$. You should choose values of $n$ that are large enough so that you are seeing only a small variation. The table values should be the average of a number of runs. You should do values of $k$ up to the point where $\gamma_k < \frac{2}{5}$.

b.) Provide your algorithmic code. (You will only need to compute the lengths of the LCS, not the string giving the LCS.)

Problem 2 (10 points):
Answer the following questions with “yes”, “no”, or “unknown, as this would resolve the P vs. NP question.” Give a brief explanation of your answer.

Define the decision version of the Interval Scheduling Problem as follows: Given a collection of intervals on a time-line, and a bound $k$, does the collection contain a subset of nonoverlapping intervals of size at least $k$?

a) Question: Is it the case that Interval Scheduling $\leq_P$ Vertex Cover?

b) Question: Is it the case that Independent Set $\leq_P$ Interval Scheduling?
Problem 3 (10 points):
Argue that the following problems are in NP. (Note: just show they are in NP, this question is not asking you to show they are NP-Complete.)

a) The *Clique Problem*. A clique in an undirected graph $G = (V, E)$ is a subset $U \subseteq V$, such that there is an edge in $E$ between every pair of vertices $x, y$ in $U$. The Clique Problem is: Given a graph $G = (V, E)$ and an integer $K$, does there exist a clique of size at least $K$.

b) The *Shortest-Paths Problem*. The formal version (as a yes-no problem): Given a directed graph $G = (V, E)$ with integer edge lengths, distinguished vertices $s$ and $t$, and an integer $K$, does there exist a path of length at most $K$ between $s$ and $t$.

b) The *4-Dimensional Matching Problem*. (Defined in Problem 4.)

d) The *Zero-Weight-Cycle Problem*. (Defined in Problem 5.)

Problem 4 (10 points):
(Kleinberg-Tardos, Page 507, Problem 7). Since the 3-Dimensional Matching Problem is NP-complete, it is natural to expect that the corresponding 4-Dimensional Matching Problem is at least as hard. Let us define 4-Dimensional Matching as follows. Given sets $W, X, Y, Z$, each of size $n$, and a collection $C$ of ordered 4-tuples of the form $(w_i, x_j, y_k, z_l)$, do there exist $n$ 4-tuples from $C$ so that no two have an element in common?
Prove that 4-Dimensional Matching is NP-Complete.

Problem 5 (10 Points):
(Kleinberg-Tardos, Page 513, Problem 17). You are given a directed graph $G = (V, E)$ with weights $w_e$ on its edges $e \in E$. The weights can be negative or positive. The Zero-Weight-Cycle Problem is to decide if there is a simple cycle in $G$ so that the sum of the edge weights on this cycle is exactly 0. Prove that the Zero-Weight-Cycle problem is NP-Complete. (Hint: Hamiltonian PATH)