Problem 1

In this problem you will show that if we don't put reasonable restrictions on the class of ensembles, average-case complexity is no easier than worst-case complexity.

(a) Show that for every $L \not\in P$ there exists an ensemble $\mu_L$ such that $(L, \mu_L)$ does not have polynomial-time heuristic algorithms. (Hint: $\mu_L$ should give a lot of weight to the "hard" instances of $L$.)

(b) Show that there exists an ensemble $\mu$ such that for every $L \in \text{NP}$, $(L, \mu)$ has polynomial-time heuristic algorithms if and only if $L \in P$. (Hint: Use the various $\mu_L$ from part (a) to construct $\mu$.)

Problem 2

In this problem you investigate the difference between polynomial-time computable and polynomial-time samplable ensembles.

(a) Let $\{G_n\}$ be a pseudorandom generator. Show that the ensemble $\mu$ obtained by choosing a random $X \in \{0, 1\}^n$ and outputting $G_n(X)$ is not polynomial-time computable. Thus if pseudorandom generators exist, then $P\text{Comp} \neq P\text{Samp}$.

(b) Show that $P\text{Comp} = P\text{Samp}$ if and only if $P = P\#P$. (Hint: For the "only if" direction, consider sampling pairs $(\varphi, a)$, where $\varphi$ is a DNF and $a$ is a satisfying assignment for $\varphi$.)
Problem 3

Show that \((L, \mu)\) has an average polynomial-time algorithm if and only if there is an algorithm \(A\) with the following properties:

- \(A\) takes two inputs \(x\) and \(\varepsilon\) and runs in time \(\text{poly}(|x|, 1/\varepsilon)\).
- For every input \(x\) and every \(\varepsilon\), \(A(x, \varepsilon)\) outputs either \(L(x)\) ("yes" if \(x \in L\), "no" if \(x \notin L\)) or the special symbol "fail".
- For every \(n\) and \(\varepsilon\), 
  \[ \Pr[A(x, \varepsilon) = "fail"] \leq \varepsilon. \]

Using this alternative definition of average polynomial-time algorithms, conclude that if \((L, \mu)\) reduces to \((L', \mu')\) and \((L', \mu')\) has an average polynomial-time algorithm, so does \((L, \mu)\).

Problem 4

An undirected graph is bipartite if it has no cycles of odd length. We define the decision problem

\[ BIPART = \{ G : G \text{ is bipartite} \}. \]

Assuming that \(USTCON \in L\), show that \(BIPART \in L\). Recall the decision problem \(USTCON\):

\[ USTCON = \{ (G, s, t) : s \text{ and } t \text{ are connected in } G \}. \]

(Hint: Look at the graph \(G^2\) whose vertices are the same as \(G\) and whose edges correspond to paths of length two in \(G\).)