In the following problems, when describing computable procedures, it is enough to give informal (but precise) descriptions of your algorithms. You definitely should not try to define Turing machines implementing these algorithms... (In particular, to describe a computably enumerable set, it is enough to give an informal algorithm for listing its elements.)

Problem 3 is a “take-home exam” problem, to be worked out individually.

1. Show that there are computably enumerable sets \( A \) and \( B \) that are \textit{computably inseparable}, meaning that \( A \cap B = \emptyset \) but there is no computable set \( C \) with \( A \subseteq C \) and \( B \cap C = \emptyset \). [Hint: We can identify a set \( C \) with its characteristic function \( \chi_C \), where \( \chi_C(n) = 0 \) if \( n \notin C \) an \( \chi_C(n) = 1 \) if \( n \in C \). Consider an effective listing \( \Phi_0, \Phi_1, \ldots \) of all partial computable functions, and for each \( e \), ensure that if \( \Phi_e \) is total then there is an \( n \) such that either \( \Phi_e(n) = 0 \) and \( n \in A \) or \( \Phi_e(n) > 0 \) and \( n \in B \).]

2. Let \( \sigma_0, \sigma_1, \ldots \) list the finite binary strings in length-lexicographic order, and say that a set of binary strings \( S \) is computable if \( \{ i : \sigma_i \in S \} \) is computable.

   A \textit{binary tree} is a set \( T \) of finite binary strings such that if \( \sigma \in T \) and \( \tau \prec \sigma \) then \( \tau \in T \). An \textit{infinite path} on a tree \( T \) is an infinite binary sequence \( \alpha \) such that every initial segment of \( \alpha \) is on \( T \).

   a. Show that every infinite binary tree has an infinite path.

   b. Show that there is a computable infinite binary tree with no computable infinite path. [Hint: Let \( \Phi_e \) be as above. If \( \Phi_e \) is total and 0,1-valued then we can think of it as an infinite binary sequence. Build your tree \( T \) so that for each such \( \Phi_e \), there is some initial segment of \( \Phi_e \) that is not on \( T \).]

   c. Let \( T \) be a computable infinite binary tree. Show that if we could compute the Halting Problem, then we could compute an infinite path on \( T \). [Hint: The s-m-n Theorem (aka the Parameter Theorem) might be helpful here.]

   d. Show that there is a computable tree \( T \) such that, if we could compute an infinite path on \( T \), then we could compute a completion of ZFC (i.e., a complete, consistent theory extending ZFC). [Note: ZFC is used here only as an example; this would work for any theory with a computable set of axioms.]
3 (take-home exam problem). Let $A$ and $B$ be computably enumerable subsets of $\mathbb{N}$. For each of the following sets, must the set be c.e.? (In each case, prove or give a counterexample.): $A \cup B$, $A \cap B$, $A \setminus B$. 