**Assignment#2; Due February 3 at start of class**

Let set **A** be non-empty recursive, **B** be re non-recursive and **C** be non-re. Using the terminology **(REC)** **recursive**, **(RE) non-recursive recursively enumerable**, **(NR)** **non-re**, categorize each set below, saying whether or not the set can be of the given category and justifying each answer. You may assume, for any set **S**, the existence of comparably hard sets   
**SE = {2x|x∈S}** and **SD = {2x+1|x∈S}**. The following is a sample of the kind of answer I require:

**Sample.) A ∩ C = { x | x ∈ A and x ∈ C }**

**REC: Yes. If A = {0} then A ∩ C = ∅ ot {0}, each of which is in REC.**

**RE: Yes. Let A = ℵE = { 2x | x ∈ ℵ }; let C = TOTD ∪ HALTE then A ∩ C = HALTE which is in RE**

**NR: Yes. If A = ℵ then A ∩ C = C, which is in NR.**

**a.) A – B = { x | x ∈ A and x ∉ B } // Set difference**

**b.) min(A, B) = { min(x , y) | x ∈ A and y ∈ B } // Minimum**

**c.) A ⊕ C = { x | x ∈ A or x ∈ C but ∉ A ∩ C } // Set exclusive union**

Be careful: Some may not be possible. If so, you must justify why this is so.

Note:

**TOT = { x | ∀ ϕx (y) ↓ }**. These are the indices of the set of algorithms.

**HALT = { <x,y> | ϕx (y) ↓ }**. This is the set of pairs of procedures and input for which the given procedure halts.

The set **SE**, for any set **S**, is defined as **{2x | x ∈ S }**

The set **SD**, for any set **S**, is defined as **{ 2x+1 | x ∈ S }.**

The complexities of **S**, **SE** and **SD** are the same. That is, all three are either in **REC**, **RE** or **NR**.