Problem 1: These are the key concepts from lecture this week:

1. Undecidability - p. 188-192 have great example proofs.
2. Reductions - p. 187-188 will help with the terminology (e.g., “reduce $A$ from $B$”, etc.)
3. Computation history - p. 192, 195, 201 give the definition and some examples.
4. Diagonalization method - p. 174-182. This concept is both elegant and difficult; make sure you understand it.
5. Mapping Reducibility - pages 206-210 (make sure you understand Theorems 5.22, 5.23, 5.28, and 5.29)
6. Rice’s Theorem

Problem 2: Show that the following language is undecidable:

$L = \{ \langle M \rangle : M$ is a Turing machine and $M$ accepts exactly the strings in $\Sigma^*$ whose length is a power of 2 $\}$.

Problem 3: Show that the following language is undecidable:

$EQ_{TM} = \{ \langle M, N \rangle : M$ and $N$ are TMs such that $L(M) = L(N) \}$

Reduce from both $ET_{TM}$ and $AT_{TM}$. Recall that $EQ_{DFA}$ was decidable.
Problem 4: Give yourself the following test, then check your answers on the back of the handout. Classify each of the following problems as either

- (D) decidable,
- (R) recognizable but not decidable,
- (C) co-recognizable but not decidable, or
- (N) neither recognizable nor co-recognizable,

and indicate which undecidable examples follow from Rice’s Theorem.

1. $EQ_{NFA}$, the Equivalence problem for NFA’s.
2. $\{\langle M \rangle \mid M \text{ is a Turing Machine that runs for at least } n \text{ steps when started with a blank input tape, where } n \text{ is the length of the string } \langle M \rangle \}$.
3. $\{\langle M \rangle \mid M \text{ is a Turing Machine that accepts at least two inputs}\}$.
4. $\{\langle M \rangle \mid M \text{ is a Turing Machine that does not halt on input } \langle M \rangle \}$.
5. $\{\langle M \rangle \mid L(M) \text{ is regular}\}$
6. $EQ_{TM}$

Problem 5: (Mapping Reducibility) Answer the following True or False:

1. $A_{TM}$ is mapping reducible to $E_{TM}$.
2. $A_{TM} \leq_{m} 0^{*}1^{*}$.

Problem 6: (Rice’s Theorem and Mapping Reducibility)
Consider the problem of testing whether a Turing machine $M$ accepts any binary string with an odd number of zeros.

1. Formulate this problem as a language; call it $ODDZ$.
2. Show that $ODDZ$ is undecidable.
**Problem 4 Solutions:**

1. D; recall the $EQ_{DFA}$ algorithm from textbook.
2. D; just simulate $M$ for up to $|\langle M \rangle|$ steps.
3. R; Undecidable by Rice’s Theorem; Recognizable by running the TM in parallel (i.e., using the dovetailing technique from class) on all input strings until it accepts two strings.
4. C; Undecidable by diagonalization. Co-recognizable by simulating $M$ on input $\langle M \rangle$ and accepting when $M$ halts.
5. N; Undecidable by Rice’s Theorem; Neither recognizable nor co-recognizable from class.
6. N; Undecidable by Rice’s Theorem; Neither recognizable nor co-recognizable from textbook.

**Problem 5 Solutions:**

1. False; $A_{TM}$ is recognizable, $E_{TM}$ is not. See Corollary 5.23.
2. False; $0^*1^*$ is decidable, $A_{TM}$ is not. See Theorem 5.22.

5: (Un)Decidability, Reducibility, Rice’s Theorem-3