Name:

This quiz is closed book and only one notes sheet. There are 100 points in seven questions and you must show your work to receive full credit. Put your name, the date, and “Exam 3” on each sheet of paper you turn in. Papers must be stapled.

1. Define a Turing Co-recognizable language. (5 points)
   My Answer: A Turing Co-recognizable language is the set of all strings in the complement of a Turing recognizable language. If $L$ is recognized by a Turing Machine, then $\overline{L}$ is as Co-recognizable language.

2. Can a single language be both Recognizable and Co-recognizable? (5 points)
   My Answer: Yes. It is very easy to construct a language $L = \{\}$ which is recognizable by a Turing machine and the language $\overline{L}$ is recognizable by a TM that halts in accept as soon as it reads the first non-blank character on the tape that is in $\Sigma$. Therefore, $L$ is the co-language for $\overline{L}$ and $\overline{L}$ is the co-language for $L$ and both are Recognizable and Co-recognizable at the same time.

3. The Halting Problem
   (a) In a few words, what is the Halting Problem? (10 points)
   My Answer: Is it possible to build a TM $D(I, TM)$ that can accept an input $I$ and a $TM$ encoded on the tape and determine if the machine will stop (not loop) on input $I$. If we can do this for any Turing machine, then the Halting Problem is decidable, i.e. computable. If not, there are problems which cannot be computed by a Turing machine.

   (b) What are the implications of the Halting Problem (and/or the Decidable Languages Problem) outside of computer programming? (10 points) (HINT: It is not “will my program ever end?”)
   My Answer: The Halting Problem can be shown not to be computable and therefore there are some problems in mathematics that cannot be computed or decided.
(c) Prove there is no TM to solve the Halting Problem (20 points)

My Answer:

Proof by Contradiction.
Assume such a TM, \( H(I, TM) \) exists and can be constructed as follows:

i. Create a universal TM \( D(I, TM) \) that reads an input and a binary encoded TM from a tape such that if \( TM \) halts on the input \( I \), then \( D(I, TM) \) halts for any \( I \) and \( TM \).

ii. Construct a TM \( H(I, TM) \) that uses \( D(I, TM) \) as a module (sub-routine, function, or method) such that if \( D(I, TM) \) halts, \( H(I, TM) \) goes into a loop. If \( D(I, TM) \) does not halt, \( H(I, TM) \) halts

iii. Run \( H(H, H) \). If \( D(H, H) \) halts, then \( H(H, H) \) does not halt. However, if \( H(H, H) \) does not halt, then \( D(H, H) \) does not halt which means \( H(H, H) \) should halt.

iv. Obviously, step 3(c)iii makes no sense, therefore \( D(I, TM) \) cannot exist nor can \( H(I, TM) \).

4. Define the following classes (5 points each):

(a) Class P

My Answer: The class of problems that can be solved by a deterministic TM in polynomial time.

(b) Class NP

My Answer: The class of problems that can be solved by a non-deterministic TM in polynomial time. The solution can be verified in polynomial time by a deterministic TM.

(c) Class NP-Complete

My Answer: The class of decision problems that are at least as hard as the hardest problems. This class is such that if we can solve on NPC problem in polynomial time with a deterministic TM in polynomial time, we can solve them all with deterministic TMs in polynomial time. By extension, we can solve all NP problems in polynomial time with a deterministic TM which implies P=NP.

5. Give an example of a problem in each of the following classes (5 points each)
(a) Class P
   My Answer: The fractional Knapsack.
(b) Class NP-Complete
   My Answer: The 0/1 Knapsack.

6. What is “the problem in computer science?” (5 points)
   My Answer: Does P=NP?

7. Given that the Bin Packing Problem is NP-Complete, what would be the steps to show that the 0/1 Knapsack is NP-Complete? (20 points)
   My Answer:
   
   (a) Cast the 0/1 Knapsack as a decision problem: Given this set of items, can I pack no less than value $k$ into a knapsack of capacity $C$?
   
   (b) Show that the 0/1 Knapsack is HP-hard (i.e, show reason to believe the best solution is $O(2^n)$ or brute force.)
   
   (c) Construct a polynomial time reduction (assuming a deterministic TM) from the Bin Packing Problem to the 0/1 Knapsack or:

   $\text{BinPacking} \leq_P \text{0/1 Knapsack}$