Fall 2024 COT 4210 Exam #3
October 31, 2024
Sheet 1: TMs and Decidability

Last Name: _____ , First Name: _____

1) (10 pts) Let L = {1} if the Lochness Monster exists and L = {0} if the Lochness Monster does not exist. Is L a decidable language? Prove your result.

2) (10 pts) Let L = { (P, n) | P is a polynomial with integer coefficients in variable x that has has at least one integer root in between 0 and n, inclusive. }

Is L a decidable language? Prove your result.

3) (15 pts) Let L = { <A> | A is a DFA that accepts all strings of the form 0^n1^n }. Is L a decidable language? Prove your result. Be as detailed as possible in your proof.

Fall 2024 COT 4210 Exam #3
October 31, 2024
Sheet 2: Countability, Undecidability

Last Name: _____ , First Name: _____

4) (15 pts) Let L = { <M, w, q> | M is a Turing Machine such that when run on input w, enters state q at least once.} Is L a decidable language? Prove your response.

5) (20 pts) Let Σ be the set of lowercase letters. To prove that Σ^* is countable, we could create a one-to-one correspondence between the positive integers and the elements of Σ^*. One such way to do this is to order the strings by length, and then break ties by alphabetical order. This list would look like: ε, a, b, c, ..., z, aa, ab, ac, ..., zz, aaa, aab, ...

Write a method in Java, getRank, that takes in a String object and returns a long, the 1-based ranking of the inputted string on this list. Some support code has been provided so that **you aren't tempted to use doubles.** Don't worry about overflow, I am only grading the theoretical aspects of your code, but do use long...

```
import java.util.*;

public class e3q5 {

     public static long pow(long b, int e) {
          long res = 1;
          for (int i=0; i<e; i++)
               res = res*b;
          return res;
     }

     public static long getRank(String s) {









     }
}
```

Fall 2024 COT 4210 Exam #3
October 31, 2024
Sheet 3: Undecidability, Reducibility

Last Name: _____ , First Name: _____

6) (15 pts) E_{LBA} = {<M>| M is an LBA and L(M) = ∅ }. We proved in class that E_{LBA} is undecidable. Sketch out the key details of the proof, highlighting the novel idea presented in this proof compared to the undecidability proofs we showed in class before it.

7) (10 pts) To show that you understand the concept of mapping reducibility, consider two languages over Σ^*, where Σ is the set of lowercase letters.

A = {odd length palindromes}
B = {even length palindromes of length 2 or greater}

To prove that language A is mapping reducible to language B, we must find a computable function f such that if $w \in A$, then $f(w) \in B$, AND if $w \notin A$, then $f(w) \notin B$. Describe in a precise way, how this function, f, that proves that A is mapping reducible to B, should work. (I would have you code it, but I did and it's just annoying. Much easier to just describe in words...) So, your function should take in an odd length string, w, and produce an even length string w', such that when w is a palindrome, so is w', and similarly, when w isn't a palindrome, w' won't be either.

8) (5 pts) In what city is the Art Institute of Chicago? _____