Homework Assignment #4
Due: Friday, March 13, 2026 at 7:00 p.m.

1. Recall that a configuration of a (deterministic, single-tape) Turing machine $M = (Q, \Sigma, \Gamma, q_0, q_{accept}, q_{reject}, \delta)$ is a snapshot of one moment of a computation performed by M. The configuration describes the contents of the tape, the state of the Turing machine and the position of the head on the tape.

[3] **(a)** Suppose that when M is run on input string x, the head never moves beyond the first m squares of the tape. Give a good upper bound B on the number of different configurations that the Turing machine may have during such a computation. State your answer in terms of $m, |Q|$ and $|\Gamma|$ and explain why your answer is correct.

[2] **(b)** Show that if M does not move beyond the first m squares of the tape during the first B steps of its computation on input x, then it never will.

[3] **(c)** Write pseudocode for an algorithm that *decides* the language
$\{\langle M, x\rangle$: when TM M runs on string x, it stays within the first $|x|^3$ squares of the tape$\}$.

2. Here, we use the notation $L(M)$ for the language of strings accepted by a Turing machine M. Let $SUB_{TM} = \{\langle M_1, M_2\rangle : M_1$ and M_2 are Turing machines and $L(M_1) \subseteq L(M_2)\}$.

[3] **(a)** Suppose you had an algorithm DECIDE-$SUB_{TM}(\langle M_1, M_2\rangle)$ that decides the language SUB_{TM}. Show how you could use this algorithm as a subroutine to design an algorithm DECIDE-$A_{TM}(\langle M_1, x\rangle)$ that decides the language A_{TM}.

[1] **(b)** Prove that SUB_{TM} is undecidable.

Bonus Question: Busy Beavers
The bonus question below is separate from the rest of Assignment 4.
The bonus question is due Monday, March 23 at 19:00 p.m.

3. In this question, we shall restrict attention to (deterministic, single-tape) Turing machines that use the tape alphabet $\Gamma = \{0, 1, 2, 3, \sqcup, \triangleright\}$. The goal is to design a Turing machine which, on input ε, takes as many steps as possible and then halts. Your machine must have exactly five states, including the initial state, the accepting state and the rejecting state.

This bonus question is worth up to 4 percent to be added to your final grade. Students who submit a correct Turing machine will get between 1 and 4 bonus points. If two students submit machines that take the same number of steps, each student will get 1 point. Excluding these students, the more steps your machine takes before halting, the more bonus points you will earn. To earn all 4 points, you will have to make your machine take a very large number of steps.

If your machine takes more than 1,000,000,000 steps on the empty input string, I may not be able to test it by running it, so you will have to provide a written proof that it does indeed halt (and that it takes lots of steps) in order to receive your bonus points.

Write your Turing machine in a YUTMFF text file named beaver.txt and submit it online at `https://webapp.eecs.yorku.ca/submit/` or by using the following command.

```
submit 2001M beaver beaver.txt
```

If your submission is syntactically incorrect or does not follow the specifications of this question (for example if it has too many states) or runs forever on the empty input string, you will receive 0 points.

For this bonus question, you may consult outside sources. A good place to start is the Computer Recreations column by A. K. Dewdney in the August, 1984 issue of *Scientific American* (pp. 19–23). It is available from the York library via `https://www-jstor-org.ezproxy.library.yorku.ca/stable/24969427`. (Note that the Turing machines discussed there use two-way infinite tapes, so you cannot use them directly to answer this question, but you might get some ideas about how to design your own busy beaver.) You should use the TM.java programme posted on the course website for Assignment 3 to test your machine (or even to help you design it).