"To loop is human; to recurse is divine." – L. Peter Deutsch

1. Construct Turing Machines that accept the following languages. You do not need to explicitly define the 7-tuple. Instead, write in clear and concise steps, as done in Examples 3.6 and 3.7 from Sipser's book.

   (a) $\{ww \mid w \in \{a, b\}^*\}$

   (b) $\{b_1\#b_2 \mid b_1, b_2 \in \{0, 1\}^*, \text{as binary numbers } b_1 > b_2\}$

   Using part (b), convince yourself that sorting of a given set of binary numbers separated by a special symbol is possible on a Turing Machine.

2. Are recursive languages closed under the following operations?

   (a) union

   (b) concatenation

   (c) complementation

   (d) star

   (e) intersection

   (f) homomorphism

   (g) inverse homomorphism

   What about the same operations for recursively enumerable languages?

3. Show that a language is decidable iff some enumerator enumerates the language in the standard order of strings, i.e. lengthwise followed by lexicographic.

4. We know that CFLs are a strict subset of recursive languages. Now, extend the model of PDAs by adding one additional stack. On reading a word we can push and pop both stacks independently of each other. Thus the transition function now looks like:

   $$\delta : Q \times \Sigma \times \Gamma \times \Gamma \rightarrow Q \times \Gamma^* \times \Gamma^*$$

   Show that 1(a) is accepted by such a machine. Attempt to characterise the class of languages accepted by such machines. What if we add even more stacks?

5. Consider the following restricted version of the halting problem: we are given a Turing Machine M, a word w and a constant c. We know that M, on any input w_0, is restricted to use at most $c\,|w_0|$ space on its tape. Does M halt on w?
   Think if this problem is recursive.