Tutorial 7, Design and Analysis of Algorithms, 2021
($T : \mathbb{N} \to \mathbb{N}$ is time-constructible if $T(n)$ can be computed by a DTM in $O(T(n))$ time)
(For simplicity, multi-tape DTM can be used in the definition of DTIME$(T(n))$)

1. (a) Design a one-tape DTM for accepting the language of palindromes:

$$L_p = \{ w \mid w \in \{0,1\}^*, w = \text{ reverse}(w) \},$$

   and find its time complexity.

   (b) Using the flipping states method, design a one-tape DTM (with only one final state) for accepting the language of non-palindromes:

$$L_{np} = \{ w \mid w \in \{0,1\}^*, w \neq \text{ reverse}(w) \},$$

   and find its time complexity.

2. Define a two dimensional TM to be a TM where its tape is an infinite grid (and the machine can move not only `Left` and `Right` but also `Up` and `Down`). Show that for every (time constructible) $T : \mathbb{N} \to \mathbb{N}$ and every Boolean function f, if f can be computed in time $T(n)$ using a two-dimensional TM then $f \in \text{DTIME}(T(n)^2)$.

3. Define a RAM TM to be a TM that has random access memory. We formalize this as follows: the machine has an infinite array A that is initialized to all blanks. It accesses this array as follows. One of the machine's work tapes is designated as the address tape. Also the machine has two special alphabet symbols denoted by R and W and an additional state we denote by q_a. Whenever the machine enters q_a, if its address tape contains $[i]_2 R$ (where $[i]_2$ denotes the binary representation of i) then the value $A[i]$ is written in the cell next to the R symbol. If its tape contains $[i]_2 W \sigma$ (where σ is some symbol in the machine's alphabet) then $A[i]$ is set to the value σ. Show that if a Boolean function f is computable within time $T(n)$ (for some time-constructible T) by a RAM TM, then it is in DTIME $(T(n)^2)$.

4. Consider the following simple programming language. It has a single infinite array A of elements in $\{0,1,B\}$ (initialized to B) and a single integer variable i. A program in this language contains a sequence of lines of the following form:
   `label :` `If` $A[i]$ `equals` σ `then cmds`.
   Where $\sigma \in \{0,1,B\}$ and `cmds` is a list of one or more of the following commands: (1) `Set` $A[i]$ `to` τ where $\tau \in \{0,1,B\}$, (2) `Goto label`, (3) `Increment` i `by one`, (4) `Decrement` i `by one`, and (5) `Output` b `and halt`, where $b \in \{0,1\}$. A program is executed on an input $x \in \{0,1\}^n$ by placing the i'th bit of x in $A[i]$ and then running the program following the obvious semantics. Prove that for every functions $f : \{0,1\}^* \to \{0,1\}$ and (time constructible) $T : \mathbb{N} \to \mathbb{N}$, if f is computable in time $T(n)$ by a program in this language, then $f \in \text{DTIME}(T(n))$.

5. Recall that normally we assume that numbers are represented as string using the *binary* basis. That is, a number n is represented by the sequence $x_0, x_1, \ldots, x_{\log n}$ such that $n = \sum_{i=0}^{\log n} x_i 2^i$, where for each $i \in [0..\log n]$ $x_i \in \{0,1\}$. However, we could have used other encoding schemes. If $n \in \mathbb{N}$ and $b \geq 2$, then the representation of n in base b, denoted by $[x]_b$ is obtained as follows: first represent n as a sequence of digits in $\{0,\ldots,b-1\}$, and then replace each digit $d \in [0..b-1]$ by its binary representation. The unary representation of n, denoted by $[n]_1$ is the string 1^n (i.e., a sequence of n ones).

(a) Show that choosing a different base of representation will make no difference to the class **P**. That is, show that for every subset S of the natural numbers, if we define $L_S^b = \{ [n]_b : n \in S \}$ then for every $b \geq 2$, $L_S^b \in \mathbf{P} \Leftrightarrow L_S^2 \in \mathbf{P}$.

(b) Show that choosing the unary representation may make a difference by showing that the following language is in **P**:

UNARYFACTORING $= \{ < [n]_1, [l]_1, [k]_1 >$: there is a prime $j \in (l, k)$ dividing $n \}$.

It is not known to be in **P** if we choose the binary representation.

6. Prove that allowing the certificate to be of size at most $p(|x|)$ (rather than equal to $p(|x|)$) in the certificate definition of NP, makes no difference. That is, show that for every polynomial-time DTM M and polynomial $p : N \to N$, the language

$$\{ x : \exists u \mid |u| \leq p(|x|) \text{ and } M(x, u) = 1 \}$$

is in NP.

7. Let `LINEQ` denote the set of satisfiable rational linear equations. That is, `LINEQ` consists of the set of all pairs (A, b) where A is an $m \times n$ rational matrix and b is an m dimensional rational vector, such that $Ax = b$ for some n-dimensional vector x. Prove that `LINEQ` is in NP.

8. Prove or disprove:

   (a) If $L_1 \in$ P and $L_2 \in$ P then $L_1 \cup L_2 \in$ P.

   (b) If $L_1 \in$ NP and $L_2 \in$ NP then $L_1 \cup L_2 \in$ NP.

9. Prove or disprove:

   (a) If $L_1 \in$ P and $L_2 \in$ P then $L_1 \cap L_2 \in$ P.

   (b) If $L_1 \in$ NP and $L_2 \in$ NP then $L_1 \cap L_2 \in$ NP.

10. For any language L, we define the language reverse(L) as the set of all strings of L in reverse order:

$$\text{reverse}(L) = \{ w \mid \text{reverse}(w) \in L \}.$$

Prove or disprove:

(a) If $L \in$ P then reverse(L) $\in$ P.

(b) If $L \in$ NP then reverse(L) $\in$ NP.