1. Let $L$ be a language over the alphabet $\Sigma = \{0, 1\}$ that is defined recursively as follows:

- $\epsilon \in L$;
- $u, v \in L \implies 0uv \in L$;
- $u, v \in L \implies 1vu \in L$.

The only strings in $L$ are the ones that can be constructed inductively by applying the above rules. Prove that $L$ is the language consisting of all strings that have an equal number of 0 symbols and 1 symbols. In other words, if $K$ denotes the language consisting of all strings with an equal number of 0’s and 1’s, then you need to show that $L = K$. To do this, you should argue $L \subseteq K$ and $K \subseteq L$.

2. A world-state in Game of Life consists of an assignment of “alive” or “dead” to each cell $(x, y)$ in the infinite 2-d grid $\mathbb{Z} \times \mathbb{Z}$. Determine, with proof, whether the set of all possible world-states is countable.

3. Consider the set of functions $f : \mathbb{N} \to \mathbb{N}$ which: (i) are increasing (meaning $f(x) > f(y)$ whenever $x > y$); and, (ii) satisfy $f(n) \leq 2n$ for all $n \in \mathbb{N}$. Determine with proof whether this set is countable.

4. One can encode the alphabet $\Sigma_1 = \{0, 1, \sqcup, \#\}$ by the alphabet $\Sigma_2 = \{0, 1\}$ by applying the map $\sqcup \rightarrow 00$, $\# \rightarrow 01$, $0 \rightarrow 10$, $1 \rightarrow 11$. Let $L \subseteq \Sigma_1^*$, and let $L' \subseteq \Sigma_2^*$ be its “encoding”. Show: $L$ is regular if and only if $L'$ is regular.

5. Consider the following languages over the alphabet $\{a, b\}$:

$$L = \{a^m b^n : n, m \in \mathbb{N}, n - m \leq 251\},$$

$$K = \{a^m b^n : n, m \in \mathbb{N}, n + m \leq 251\}.$$

Show that one of these languages is regular and the other is not. For the proof of regularity, all you need to do is draw or describe a DFA; a proof of correctness is not required. For the proof of irregularity, present all the details.

6. Let’s say that a DFA $D$ is “interesting” if it accepts at least one string $x$ with $k \leq |x| \leq 2k$, where $k$ denotes the number of states in $D$.

(a) Show that $L = \{(D) : D$ is interesting$\}$ is decidable.

(b) Show that $D$ is interesting if and only if $D$ accepts infinitely many strings.

7. Fix $\Sigma = \{a, b\}$. Given a word $w$, we let filter($w$) denote $w$ with its even-indexed characters removed. For example, filter($a$) = $a$, filter($ab$) = $a$, filter($abba$) = $ab$, and filter($\epsilon$) = $\epsilon$. For a language $L \subseteq \Sigma^*$, define

$$\text{FILTER}(L) = \{\text{filter}(x) : x \in L\}.$$ 

Show that if $L$ is regular, then so is $\text{FILTER}(L)$.
8. Fix $\Sigma = \{a, b\}$. For a language $L \subseteq \Sigma^*$, define

$$\text{DOUBLE}(L) = \{x \in \Sigma^* : \text{filter}(x) \in L\}.$$ 

Show that if $L$ is regular, then so is $\text{DOUBLE}(L)$. 

9. Consider the following TM called $M$, which has input alphabet $\{a\}$ and tape alphabet $\{a, \sqcup\}$. 

![Diagram of TM]

Prove that $M$ does not halt on input $aaa$. Your proof should use the notion of a configuration. 

10. (a) We say that a language $L$ is acceptable if there exists a Turing Machine $M$ such that $M(x)$ accepts for all $x \in L$ and $M(x)$ either rejects or loops for all $x \not\in L$. Show that $L$ is decidable if and only if $L$ and $\overline{L} = \Sigma^* \setminus L$ are both acceptable. 

(b) Recall that the language $\text{HALTS} = \{(M, x) : M \text{ is a TM and } M(x) \text{ halts}\}$ is undecidable. Show that $\text{HALTS}$ is acceptable (and that therefore, the decidable languages are a strict subset of the recognizable languages).

(c) Show that $\text{HALTS}$ is not acceptable.

(d) Show that $\{(M) : M \text{ is a TM and } \exists x \in \Sigma^* \text{ such that } M(x) \text{ halts}\}$ is acceptable.

(e) Show that every acceptable language $L$ is reducible to $\text{HALTS}$. 

11. (a) Fix a Turing Machine $M$ with input alphabet $\Sigma$ and consider the language $L_M = \{x \in \Sigma^* : M(x) \text{ halts}\}$. Is there an $M$ such that $L_M$ is decidable? 

(b) Fix a string $x \in \Sigma^*$ and consider the language $L_x = \{(M) : M(x) \text{ halts}\}$. Is there an $x$ such that $L_x$ is decidable? 

12. Recall that for $w \in \Sigma^*$, $w^R$ denotes the reversal of $w$. For example $10111^R = 11101$. Show that 

$$K = \{(M) : M \text{ is a TM which accepts } \langle M \rangle^R\}$$

is undecidable.

13. Let $K$ be the following language:

$$K = \{(M) : M \text{ is a TM and } L(M) \text{ is finite or } L(M) = \Sigma^*\}.$$ 

Prove that $K$ is undecidable.
14. Prove or give a counter-example to the following claim: Given any two functions \( f(n) \) and \( g(n) \), either \( f(n) = O(g(n)) \) or \( g(n) = O(f(n)) \).

15. (a) Draw a Turing Machine \( M \) which decides the language \( \{0^n1^m : n, m \in \mathbb{N}\} \).
   (b) Determine the running time function \( T_M(n) \) completely exactly.
   (c) Prove that \( T_M(n) = \Theta(n) \).
   (d) Although we rarely consider “best-case running time”, let’s do it in this problem. Define
   \[ U_M(n) = \min_{\text{instances } x \text{ of length } n} \{ \# \text{ of steps } M \text{ takes on } x \}. \]
   Determine the function \( U_M(n) \) exactly. (Probably it will involve “cases”.)
   (e) Prove that \( U_M(n) = \Theta(1) \).

16. Let \( T(n) \) satisfy the following recurrence relation:
   \[ T(1) = c, \quad T(n) \leq 3 \cdot T(n/5) + k \cdot n^4 \quad \text{for } n > 1, \]
   where \( c \) and \( k \) are some constants that don’t depend on \( n \). You can assume \( n \) is power of 5, i.e. \( n = 5^t \) for some \( t \in \mathbb{N} \).
   (a) Consider the recursion tree corresponding to the above recursive relation. Determine the total number of nodes in the tree, in terms of \( n \), using the \( \Theta(\cdot) \) notation. Prove your claim.
   (b) Prove a tight upper bound on \( T(n) \) using the big-O notation.

17. Describe a “linear-time reduction from multiplication to squaring”. That is, suppose you are given access to black-box that, given a number \( B \), returns \( B^2 \) to you. Show how to multiply two \( n \)-bit numbers using time \( O(n) \) plus at most a constant (like, one or two or three) number of calls to the squaring black-box.
   (The point of this problem is to illustrate that if you didn’t know an algorithm for doing faster-than-quadratic-time multiplication, and you were trying to discover such an algorithm, you could WLOG focus just on doing the special case of faster-than-quadratic-squaring. In fact, both Kolmogorov and Karatsuba knew this fact, and it helped Karatsuba discover his algorithm for multiplying two \( n \)-bit numbers in time \( O(n^{1.58}) \).)

18. Fix some \( 0 < \epsilon < 1 \). Design a cake cutting algorithm for \( n \) players that finds an allocation \( (A_1, \ldots, A_n) \) such that for all \( i = 1, \ldots, n-1 \) (all players except \( n \)), \( 0 < V_i(A_i) \leq \epsilon \). Analyze the complexity of your algorithm in the Robertson-Webb model. You may assume that for any two distinct points \( x, y \in [0,1] \), and any player \( i \in N \), \( V_i([x,y]) > 0 \), that is, each player has a strictly positive value for any interval that is not a single point.