Assignment 1

Exercise 1.7

(a) $O(T(n) \times T(n)) = O(T^2(n))$

(b) Note that since we do not know how large the grid will be in each dimension in the end, we don't want to store it row-by-row or column-by-column. So, we store it in *diagonals*, i.e., $(1,1)$, $(1,2)$, $(2,1)$, $(1,3)$, $(2,2)$, $(3,1), \ldots$ Then the worst delay incurred by this organization is when going from $(1,1)$ to $(T(n), T(n))$, which now takes $O(1+2+3+4+\ldots+2T(n)) = O(T^2(n))$ time instead of $2T(n)$ time it takes in the grid TM.

Exercise 1.10

We build a TM that computes $f(n)$ in $O(T(n)$ time. If we think of A as the infinite working tape of the TM we build, then i is the current head position. Also note that the number of lines of code in the program is *fixed* (we are not required to design a TM that executes *all* programs, just a given) one). So, we encode the program line

$$label : \texttt{If } A[i] \texttt{ equals } \sigma \texttt{ then } cmds$$

into a state $(label, \sigma)$ for every *label* and σ (note that both are *fixed* for a particular program line). The transition function δ is well-defined, since given the current state $(label, \sigma)$ and head position i, we know the tape contents $A[i]$, what the head writes at $A[i]$, where its next position is, and what the new state/program line is.

Exercise 1.15

(a) The input size shrinks by a factor of $\log b$ when written base-b. So, if the running time is $O(n^c)$ when the input x is written in binary and $|x| = n$, then $O((\frac{n}{\log b})^c) = O(n^c)$ is the running time when x is written base-b.

(b) The following algorithm solves the problem: Reject $\langle n, l, k \rangle$ if $l > k$, or $l > n$ or $k > n$. For every number $l + 1 \leq m \leq k - 1$, first check whether $n = 0(mod\ m)$. If yes, then for every number $2 \leq q \leq \lceil \sqrt{m} \rceil$, check whether $m = 0(mod\ q)$. If none of these q's divides m (i.e., m is prime), then accept $\langle n, l, k \rangle$. If there is no prime m divides n, then reject $\langle n, l, k \rangle$.

Assuming that each arithmetic operation (including $\sqrt{\ }$) takes time $\log n$, using the binary representation of all numbers (note that all are at most n in value) and the elementary algorithm for it, then the running time of the algorithm is $O(n^{3/2} \log n)$.

Exercise 2.9

Given exercise 2.8, then any $L \in NP$ and $L' = HALT$ show that $\leq_R$ is not symmetric. (To solve 2.8, note that we can construct a TM M that given a $3SAT$ formula ϕ, it goes over all truth assignments, and if none satisfies the formula then goes into an infinite loop. Then $HALT(M, \phi) = 1$ iff $\phi \in 3SAT$.)

Exercise 2.10

Let M_1, M_2 be the TMs in the definitions of L_1, L_2 respectively, i.e.,

$$\begin{aligned} x &\in L_1 \Leftrightarrow \exists u_1 : M_1(x, u_1) = 1 \\ x &\in L_2 \Leftrightarrow \exists u_2 : M_2(x, u_2) = 1 \end{aligned}$$

Define TM $M_\cup(x, Qu)$ so that it just runs $M_Q(x, u)$ for $Q \in \{1, 2\}$. Then

$$x \in L_1 \cup L_2 \Leftrightarrow \exists Qu : M_\cup(x, Qu) = 1$$

Similarly, define TM $M_\cap(x, u_1|u_2)$ so that it accepts iff $M_1(x, u_1) = 1 \wedge M_2(x, u_2) = 1$. Then

$$x \in L_1 \cap L_2 \Leftrightarrow \exists u_1|u_2 : M_\cap(x, u_1|u_2) = 1$$

Exercise 2.15

$VERTEX\ COVER(G, k)$ reduces to $INDSET(G, n - k)$, and $CLIQUE(G, k)$ reduces to $INDSET(\bar{G}, k)$.

Exercise 2.32

(a) Input size $|x| = \log x$ in binary, and $|x| = x$ in unary

(b) Let M be the polynomial time NDTM that on input x decides L in $2^{p(\log x)}$ time. Let M' be the NDTM that takes input $x = [w]_{unary}1^{2^{p(\log w)}}$ of size $|x| = w + 2^{p(\log w)}$ and (i) strips it of the last $1^{2^{p(\log w)}}$ 1's (in time $O(1^{2^{p(\log w)}}) = O(|x|)$, i.e., *linear*), (ii) writes w in binary (in time $O(w) = O(|x|)$, i.e., again linear), and (iii) runs $M([w]_{binary})$ in time $O(2^{p(\log w)}) = O(|x|)$ (again linear). Then M' decides L' in linear time, i.e., $L' \in NP$.

(c) Since L' is a unary language in NP, then $L' \in P$, i.e., there is a *deterministic* TM M'' that decides whether input $x = [w]_{unary}1^{2^{p(\log w)}} \in L'$ in $q(|x|)$ time for some polynomial q. Since $x \in L' \Leftrightarrow w \in L$, we can decide whether some $w \in L$, by constructing $x = [w]_{unary}1^{2^{p(\log w)}}$ in $O(2^{p(\log w)})$ time, and then run $M''(x)$ in $O(q(|x|)) = O(q(2^{p(\log w)})) = O(2^{p'(\log w)})$ time for some polynomial p'. Therefore we can deterministically decide L in $O(2^{p'(\log w)})$ time, i.e., $L \in EXP$, i.e., $NEXP \subseteq EXP$ (the other direction is trivial).