CSC4010: Introduction to Theoretical Computer Science
Problem Set 5

Recap

- $\mathrm{TIME}(T(n))$ is the class of languages decidable in $O(T(n))$ time, and $\mathsf{P} = \bigcup_{k \geq 1} \mathrm{TIME}(n^k)$.
- A language is in NP if it has polynomial-length certificates checked by a polynomial-time verifier: every yes-instance has an accepting certificate, and no certificate makes the verifier accept a no-instance.
- $A \leq_{\mathrm{P}} B$ means that there is a polynomial-time computable function f such that $x \in A \iff f(x) \in B$ for every input x.

1. Determine whether each statement is true or false, and give a short explanation.
   (a) If a Turing machine decides a language L with worst-case running time $n^2+10n+100$, then $L \in \mathrm{TIME}(n^3)$.
   (b) If a Turing machine decides a language L with worst-case running time $2^n + 10n + 100$, then $L \notin \mathsf{P}$.
   (c) There exists a decidable language that does not belong to $\mathrm{TIME}(2^n)$.
   (d) For every pair of functions $f, g : \mathbb{N} \to \mathbb{N}$ such that $f(n) \log f(n) = o(g(n))$ and g is computable, $\mathrm{TIME}(f(n)) \subsetneq \mathrm{TIME}(g(n))$.
   (e) If $A \leq_{\mathrm{P}} B$ and $A \in \mathsf{P}$, then $B \in \mathsf{P}$.

2. Consider the language
$$L = \big\{ w \in \{0,1\}^* : |w| = 2^k \text{ for some integer } k \geq 0 \big\}.$$
   (a) Give a high-level description of a Turing machine that decides L. You may use either a single tape or multiple tapes.
   (b) Determine a tight asymptotic bound on the worst-case running time of your machine, using Θ-notation.

3. Let $G = (V, E)$ be a finite, simple, undirected graph. An *independent set* is a set of vertices in which no pair of distinct vertices is joined by an edge.

   For positive integers k, define
$$\text{INDEPENDENT-SET} = \big\{ \langle G, k \rangle : G \text{ has an independent set of size } k \big\}.$$
   (a) Show that $\text{INDEPENDENT-SET} \in \mathsf{NP}$ by describing certificates and a polynomial-time verifier. Explain why your verifier is correct and why the certificate length and running time are polynomial in the input length.
   (b) Show that $\text{INDEPENDENT-SET} \in \mathsf{NP}$ using a polynomial-time reduction to CLIQUE. Describe the reduction, prove its correctness, and explain why it is computable in polynomial time and establishes the desired conclusion. You may use the fact that $\text{CLIQUE} \in \mathsf{NP}$.

Solutions For Problem Set 5

1. (a) **True.** Since $n^2 + 10n + 100 = O(n^3)$, the given machine establishes $L \in \mathrm{TIME}(n^3)$.

   (b) **False.** A slow machine for L does not rule out a polynomial-time decider for the same language.

   (c) **True.** Since 4^n is time-constructible and $2^n \log(2^n) = o(4^n)$, the Time Hierarchy Theorem gives

   $$\mathrm{TIME}(2^n) \subsetneq \mathrm{TIME}(4^n).$$

   A language in the difference is decidable but not in $\mathrm{TIME}(2^n)$.

   (d) **False.** The Gap Theorem gives a computable $h(n) \geq n$ such that $\mathrm{TIME}(h(n)) = \mathrm{TIME}(2^{h(n)})$. Taking $f = h$ and $g = 2^h$, we have $f \log f = o(g)$ and g computable, but the time classes are equal.

   Another example is $h(n) = \max\{1, \lfloor \frac{1}{2} \log_2(n+1) \rfloor\}$. Again take $f = h$ and $g = 2^h$. Then $g(n) = \Theta(\sqrt{n})$ and $f \log f = o(g)$, but $\mathrm{TIME}(f) = \mathrm{TIME}(g) = \mathrm{TIME}(1)$. Indeed, on a standard sequential-input TM, an $o(n)$ worst-case time bound implies constant time: choose a fixed length N on which every run halts before reaching the input's end. Each run is unchanged on extensions of its input, giving a constant bound on all longer inputs.

   (e) **False.** For example, $\varnothing \leq_{\mathrm{P}} A_{\mathrm{TM}}$ by mapping every input to a fixed no-instance of A_{TM}. Yet $\varnothing \in \mathsf{P}$ and A_{TM} is undecidable. The valid implication is $B \in \mathsf{P} \Rightarrow A \in \mathsf{P}$.

2. (a) Reject the empty input. In both constructions below, each full pass accepts if there was exactly one remaining symbol at its start, rejects if there was an odd number greater than one, and otherwise repeats.

   Two tapes. Starting with the original input on one tape, scan and erase the current block while writing one 1 for each pair of symbols onto the other tape, forming a contiguous block. After each continuing pass, reset both heads and swap the tape roles.

   One tape. Repeatedly scan the entire original input interval, ignoring crossed-out symbols and crossing out every second remaining symbol. Each continuing pass in either construction halves the number of symbols, so both machines accept exactly the positive power-of-two lengths.

   (b) Let $n = |w|$. For the two-tape machine, a round on m symbols, including returning both heads, takes $O(m)$ time. Hence

   $$T(n) = O\Big(n + \frac{n}{2} + \frac{n}{4} + \cdots\Big) = O(n).$$

   Its first round scans all n input symbols, so $T(n) = \Theta(n)$.

   For the one-tape machine, each pass takes $\Theta(n)$ time because crossed-out cells remain in place. There are $\nu_2(n)$ halving passes and one final check, where $\nu_2(n)$ is the largest r such that 2^r divides n. Therefore, for $n \geq 1$,

   $$T(n) = \Theta\big(n(1 + \nu_2(n))\big).$$

   In particular, this is $\Theta(n \log n)$ when n is a power of two, but only $\Theta(n)$ when n is odd.

3. (a) Use a set $S \subseteq V$ of k vertices as the certificate. The verifier checks its size and that no two vertices in S are adjacent; it accepts exactly the independent sets of size k. The certificate lists at most $|V|$ vertices, and verification examines at most $|V|^2$ pairs, so both bounds are polynomial in the input length.

   (b) Map $\langle G, k \rangle$ to $\langle \overline{G}, k \rangle$, where $\overline{G}$ has the same vertices and an edge between distinct vertices exactly when G does not. Then

   $$G \text{ has an independent set of size } k \iff \overline{G} \text{ has a clique of size } k.$$

   The complement graph is constructed in polynomial time by examining all pairs of vertices. Compute this reduction and then use the CLIQUE verifier. Since $\mathrm{CLIQUE} \in \mathsf{NP}$, $\mathrm{INDEPENDENT\text{-}SET} \in \mathsf{NP}$.