CS 332: *Theory of Computation* — Prof. Mark Bun
Boston University — April 8, 2025

Homework 9 – Due Tuesday, April 15, 2024 at 11:59 PM

Reminder Collaboration is permitted, but you must write the solutions *by yourself without assistance*, and be ready to explain them orally to the course staff if asked. You must also identify your collaborators and write "Collaborators: none" if you worked by yourself. Getting solutions from outside sources such as the Web or students not enrolled in the class is strictly forbidden.

Note You may use various generalizations of the Turing machine model we have seen in class, such as TMs with two-way infinite tapes, stay-put, or multiple tapes. If you choose to use such a generalization, state clearly and precisely what model you are using.

Problems There are 3 required problems.

1. (**Logs and Asymptotic Notation**) Use the formal definitions of O and o notation to prove the following statements.

   (a) Let x, y, z be variables representing non-negative numbers. Simplify the following expression so it is of the form $a \log_2 x + b \log_2 y + c \log_2 z$, where a, b, c are constants: $\log_2\left(\frac{\sqrt{x}\cdot y}{z^2}\right) + \log_4(16^{\log_2 y})$.

   (b) Prove that $n^2(3\log_7 n + n) = O(n^3)$ by showing that there exists a constant $c > 0$ and a natural number n_0 such that $n^2(3\log_7 n + n) \le cn^3$ for every $n \ge n_0$. (Hint: You can use without proof the fact that $\log_2 n \le n$ for every $n \ge 1$.)

   (c) Prove that $3n = o(n^2)$ by showing that for every constant $c > 0$, there exists a natural number n_0 such that $3n \le cn^2$ for every $n \ge n_0$.

   (d) Prove that $3^{\sqrt{n}} = 2^{o(n)}$ by using the fact that $f(n) = o(g(n))$ if $\lim_{n\to\infty} f(n)/g(n) = 0$.

2. (**Polynomial-Time Algorithms**)

   (a) Let $A = \{0^{3^m} \mid m \ge 0\}$. Show that $A \in \mathsf{TIME}(n \log n)$ and $A \in \mathsf{SPACE}(n)$ by i) giving an **implementation-level** description of a basic, single-tape Turing machine M that decides A, ii) briefly explain why your TM correctly decides A, and iii) analyzing the running time and space usage of M.

   (b) An undirected graph $G = (V, E)$ is *triangle-free* if for every triple of vertices u, v, w, it is **not** the case that $(u, v), (v, w)$, and (w, u) are all edges in the graph. Let $TF = \{\langle G\rangle \mid G \text{ is triangle-free}\}$. Show that $TF \in \mathsf{P}$ by i) giving a high-level description of a polynomial-time algorithm deciding TF, ii) analyzing the correctness of your algorithm, and iii) explaining why your algorithm runs in polynomial time.

   You don't need to specify the exact polynomial runtime that your algorithm runs in, since this may depend on implementation details that are suppressed in a high-level description. Just give a convincing argument that the runtime is polynomial as in the examples in Chapter 7.2 of Sipser.

   (c) The *Fibonacci sequence* is defined by the following recurrence: $F_0 = 0, F_1 = 1$, and $F_n = F_{n-1} + F_{n-2}$ for every $n \ge 2$. Prove that there is **no** polynomial-time algorithm that takes as input a natural number n (written in binary) and outputs (i.e., writes to its tape) the number F_n (again, written in binary).

Hint: How big can the numbers F_n get as a function of n?

(d) Give a high-level description of a polynomial-time algorithm that takes as input a natural number n (written in **unary**, i.e., as the string 1^n) and outputs the number F_n (written in **binary**). Explain why your algorithm is correct and why it runs in polynomial time.

Hint: You can use without proof the fact that Turing machines can perform basic arithmetic operations on binary numbers, like addition, in polynomial time.

(e) Show that the complexity class P is closed under the union operation.

3. (**Hierarchy Theorems**) For this problem, you can assume without proof that any reasonable-looking function is time-constructible.

   (a) Show that $\mathsf{P} \subseteq \mathsf{TIME}(2^n)$.

   (b) Use the time hierarchy theorem to show that $\mathsf{EXP} \not\subseteq \mathsf{TIME}(2^n)$.

   (c) Combine parts (a) and (b) to conclude that $\mathsf{P} \neq \mathsf{EXP}$.