CS 332: *Theory of Computation*
Boston University

Professor Mark Bun
April 6, 2020

Homework 7 – Due Monday, April 13, 2020 before 2:00PM

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. Getting solutions from outside sources such as the Web or students not enrolled in the class is strictly forbidden.

Exercises Please practice on the following exercises and solved problems in Chapter 7: 7.1, 7.2, 7.4, 7.6, 7.8–7.11. The material they cover may appear on exams.

Note If you need to describe a Turing machine, please give a high-level description as in Chapter 4.1 of Sipser or in Lecture 13.

Problems There are 3 mandatory problems.

1. (**Review of asymptotic notation**) This problem will be graded automatically by Gradescope. Please enter your answers manually by completing the assignment Homework 7-Problem 1. For each of the following, select *true* or *false* using the radio buttons on Gradescope.

   (a) $2^{10} = O(n)$

   (b) $16n = O(n)$

   (c) $n^4 = O(n^2 \log n)$

   (d) $n \log n + 10n = O(n^2)$

   (e) $3^n = O(2^n)$

   (f) $3^n = 2^{O(n)}$

   (g) $2^{2^n} = O(2^{2n})$

   (h) $n^n = O(n!)$

   (i) $n = o(n)$

   (j) $2n = o(n^2)$

   (k) $2^n = o(3^n)$

   (l) $1 = o(n)$

   (m) $2 \log n = o(\log n)$

   (n) $\frac{1}{3} = o(1)$

   (o) $\log_2 n = \Theta(\log_3 n)$

   (p) $2^n = \Theta(4^n)$

   (q) $n^5 = \Theta(32^{\log_2 n})$

   (r) $n^3 = \Omega(n^3)$

   (s) $\log n = \Omega(\log(\log n))$

   (t) $2^{5^n} = \Omega(5^{2^n})$

2. (**Exponentiation cipher**) An exponentiation cipher encodes a message A using a ciphertext $C = A^e \pmod{p}$ where p is a prime number and e is an integer exponent. (Here A and C are also integers.) You are given integers A, C, e and p, and you would like to determine whether C is a valid ciphertext for message A.

(a) Formulate this problem as a language EC.

(b) Explain why the following algorithm for EC does not run in polynomial time: *Compute A^e using $e-1$ multiplications. Take the result modulo p using one integer division, and compare the answer to C.*

(c) Show that $EC \in \mathrm{P}$. Analyze the running time of your algorithm using O-notation.

Hint: First, find an algorithm for the case when e is a power of 2.

3. (**Closure properties of P**) For both parts of this problem, analyze the running time of your algorithms using O-notation. Prove that P is closed under

(a) concatenation;

(b) star.

Hint: Use dynamic programming. Let $A \in \mathrm{P}$. On input $y = y_1 \cdots y_n$ for $y_i \in \Sigma$, build a table indicating for each $i \leq j$ whether the substring $y_i \cdots y_j \in A^*$.