University of Nevada, Las Vegas Computer Science 456/656 Spring 2021

Answers to Assignment 6: Due Thursday April 22, 2021

1. Consider the following well-known complexity classes.

$$\mathcal{NC} \subseteq \mathcal{P} - \text{TIME} \subseteq \mathcal{NP} - \text{TIME} \subseteq \mathcal{P} - \text{SPACE} \subseteq \mathbf{EXP} - \text{TIME} \subseteq \mathbf{EXP} - \text{SPACE}$$

The *mover's problem* is, given a room with a door and pieces of furniture of various shapes and sizes, can the furniture be moved into the room through the door?

The *crane operator's problem* is, given a room and pieces of furniture of various shapes and sizes, can the furniture be placed into the room after the roof is removed?

For both furniture problems, we assume that no piece of furniture can ever be fully or partially on top of another.

(a) Which of the above complexity classes is the smallest class which is known to contain the mover's problem?

$\mathcal{P}$-SPACE

(b) Which of the above complexity classes is the smallest class which is known to contain the crane operator's problem?

$\mathcal{NP}$

(c) Which of the above complexity classes is the smallest class which is known to contain the context-free grammar membership problem?

$\mathcal{P}$-TIME

(d) Which of the above complexity classes is the smallest class which is known to contain the circuit valuation problem, which is the problem of determining the output of a Boolean circuit with given inputs?

$\mathcal{P}$-TIME

(e) *Generalized checkers* is the game of checkers played on an $n \times n$ board. (The standard game uses an 8×8 board.) Which of the above complexity classes is the smallest class which is known to contain the problem of determining whether the first player to move, from a given configuration, can win?

EXP–TIME

2. We do not know for sure that the complexity classes given in problem 1 are all distinct, even though it is "generally believed" that they are. However, we do know, for sure, that they are not all equal. Name two of these classes which are known to be different.

$\mathcal{P}$-TIME is a strict subclass of EXP-TIME by the time hierarchy theorem, and $\mathcal{P}$-SPACE is a strict subclass of EXP-SPACE by the space hierarchy theorem.

3. Roughly speaking, f is a *one-way function* if $f(x)$ can be computed in polynomial time for any string x, but there is no polynomial time randomized algorithm which can invert f; that is, given $f(x)$, find, with high probability, a string x' such that $f(x) = f(x')$. Such a function would be useful in cryptography.

   The formal definition is given at `https://en.wikipedia.org/wiki/One-way_function` There are some functions that are generally believed to be one-way, but no one knows for sure. Prove that if $\mathcal{P} = \mathcal{NP}$, no one-way function exists.

   Suppose $\mathcal{P} = \mathcal{NP}$, and f is a one-way function. Given the string $f(x)$, for a string x of length n, it is possible to find x using a non-deterministic machine within $O(n^k)$ steps for some constant k, by making all the right guesses. If $\mathcal{P} = \mathcal{NP}$ that computation can be emulated by a deterministic machine in polynomial time. By definition. f is then not a one-way function.

4. The *binary integer factorization* problem is, given a binary numeral for an integer n, and another "benchmark" numeral for an integer a, determine whether n has a factor greater than 1 and less than a.

   (a) Prove that the binary integer factorization problem is in $\mathcal{NP}$. (It is also in co-$\mathcal{NP}$, but that is harder to prove.)

   If there is a factor of n which is more than 1 and less than a, that factor itself is a certificate.

   (b) Show that if integer factorization is $\mathcal{P}$--TIME RSA encryption can be broken in polynomial time. (You don't have to write all the details: a sentence or two will suffice.)

   The public key for RSA incryption is a large integer n, written in binary, which is the product of two large primes, p and q. Anyone who knows both n and p can decrypt a message in polynomial time. If the binary integer factorization problem is polynomial then anyone can find the factors of n, hence the encryption can be broken.