Undergraduate Complexity Theory
Lecture 1: Course Overview

Marcythm

July 8, 2022

1 Lecture Notes

The main problem: What is the most "efficient" way to solve any computational task?

Efficient with what resources?

1. time
2. space / memory
3. data structure accesses / interactions
4. random bits

e.g. the PATH problem

1. Find a "efficient" algorithm for PATH (451)
2. Prove that there's no faster algorithm (455)

Major tool: reductions

Remark 1.1. Decrease the number of questions, without increasing the number of answers.

Many open problems:

1. $\mathsf{P} = \mathsf{NP}$?: Is finding a solution as fast as recognizing one?
2. $\mathsf{P} = \mathsf{NC}$?: Is every algorithm efficiently parallelizable?
3. $\mathsf{P} = \mathsf{L}$?: Do algorithms ever need to allocate memory?
4. $\mathsf{P} = \mathsf{PSPACE}$?: Solvable w/o much memory implies solvable w/o much time?
5. $\mathsf{P} = \mathsf{BPP}$?: Can every efficient randomized algorithm be made deterministic?
6. $\mathsf{P} = \mathsf{QuasiLIN}$?: Can every "efficient" algorithm be made actually efficient? FALSE!

Definition 1.2. *Alphabet* Σ is a finite nonempty set of **symbols**. e.g. $\Sigma = \{0, 1\}$.

Definition 1.3. Σ^n: strings of length exactly n. e.g. $\Sigma^2 = \{00, 01, 10, 11\}$.

Definition 1.4. $\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \cdots = \bigcup_{n \geq 0} \Sigma^n$ is the set of all finite-length strings.

Definition 1.5. $\langle X \rangle_\Sigma$ denotes a fixed "reasonable" encoding of mathematical object X by a string in Σ^*.

How to encode X in unary representation? $\langle X \rangle_{\{1\}} = \langle 1 \langle X \rangle_{\{0,1\}} \rangle_{\{1\}}$.

"Reasonable":

1. if $X \neq Y$ then $\langle X \rangle \neq \langle Y \rangle$.
2. "easy" conversion between $\langle X \rangle$ and data structure representing X.
3. $\langle X \rangle$ shouldn't be "much" longer than it "needs to be".

TECHNICALLY we assume that every string corresponds to some object. (invalid string $\rightarrow$ default obj.)

Three kinds of computational problems:

1. Decision problem: a function $\Sigma^* \rightarrow \{\text{yes, no}\}$. e.g. IsPrime.
2. Function problem. input string, output string that is "correct" answer, always only one answer. e.g. PrimeFactorization.
3. Search problem: Similar to function problem but the number of answer is unrestricted. e.g. FindPath: given $\langle G, s, t \rangle$, output $\langle$a path from s to $t\rangle$ or $\langle$no such path exists$\rangle$.

We mainly (WLOG) work with decision problems.

2 Reading

2.1 Sipser 0.2 (Mathematical Notions and Terminology)