24158

Computational Complexity

Introduction

Olga Tveretina

University of Karlsruhe/KIT
olga@ira.uka.de

21.10.2009

Textbooks

(1) Sanjeev Aurora and Boaz Barak. *Complexity Theory: A Modern Approach*.
http://www.cs.princeton.edu/theory/complexity

(2) Michael Sipser. *Introduction to the Theory of Computaion*.

(3) Oded Goldreich. *Computational Complexity: A Conceptual Perspective*.
http://www.wisdom.weizmann.ac.il/∼oded/cc-book.html

Computational complexity

- Focuses on classifying computational problems according to their inherent difficulty.
- A problem is regarded as inherently difficult if solving the problem requires a large amount of resources, independent of the used algorithm.
- Considers mathematical models of computation and the amount of resources needed to solve them, such as time and storage.

Closely related fields

- Analysis of algorithms: To determine the amount of resources (such as time and storage) necessary to execute them.
- Computability theory: For which decision problems do algorithms exist.
- Computational complexity theory: For which decision problems do efficient algorithms exist.

This raises the questions:

- What ‘resources’ do we wish to be employed ‘efficiently’
- What do we mean by ‘efficient’?

Computational problems

- Multiplication: Given two integer numbers a and b, compute their product $a \cdot b$.
- Dinner party problem: Given a list of acquaintances and a list of containing all pairs of individuals who are not on speaking terms with each other, find the largest set of acquaintances you can invite to a dinner party such that you do not invite any two who are not on speaking terms.

Efficiency of multiplication

- Repeated addition: add a to itself $b - 1$ times.
- Grade-school algorithm:

$$\begin{array}{r} 12 \\ \underline{23} \\ 36 \\ \underline{24} \\ 276 \end{array}$$

Hence, for multiplying two n-digit numbers:

- The repeated addition uses $n \cdot 10^{n-1}$ additions.
- Thee grade-school algorithm uses $2n^2$ additions.
- The fastest known algorithm the Fast Fourier Transform uses $(c \cdot n \cdot \ln n \cdot \ln \ln n)$ operations.

Efficiency of solving the dinner party problem

- Obvious inefficient algorithm: Try all possible subsets from the largest to the smallest, and stop after a subset that does not include any pair who do not get along.
- Running time for n people = the number of subsets = 2^n.
- To organize a 70-person party, supercomputers would spend thousands of years .
- Surprisingly: We still do not know significantly better algorithms!

Representing problem instances

- When considering computational problems, a problem instance is a string over an alphabet.
- Integers can be represented in binary notation, graphs can be encoded via their adjacency matrices.
- The independent of the choice of encoding can be achieved by ensuring that different representations can be transformed into each other efficiently.

Upper and lower bounds on the complexity of problems

- Proving upper and lower bounds on the minimum amount of time required by the most efficient algorithm solving the problem.
- The running time of a particular algorithm is measured as a function of the length $|x|$ of the input x.

Interesting questions about computational efficiency

- Do some tasks inherently require exhaustive search? (P vs NP)
- Can algorithms use randomness to speed up computation?
- Can problems be solved quicker if only approximate solutions are required?
- Can we use computationally hard problems, to construct, fo example, cryptographic protocols that are unbreakable?
- Can we use quantum mechanical properties to build faster computers?
- Can we generate mathematical proofs automatically?