ROCKY MOUNTAIN
JOURNAL OF ffiATHEMATICS
Volume 17, Number 1, Winter 1987

COMPLEXITY OF COMPUTER ALGORITHMS

RONALD J. LEACH

Introduction. This paper is intended to provide an introduction to the study of complexity of computer algorithms. No special knowledge of computers is needed; the emphasis is on the ideas involved in algorithms and not on any special features of any particular computer or computer language. Our viewpoint is that a computer is a large, dumb machine capable of doing arithmetic and comparisons at an extremely rapid rate. It performs these operations according to a precisely given set of instructions called a computer program.

We will not attempt to survey the entire field of algorithm complexity in this paper; instead we will concentrate on a few algorithms indicating some current problems and directions in computer science and related problems in mathematics.

The paper is divided into four parts. Part one contains a very brief discussion of von-Neumann's model of a computer (as a sequential rather than a parallel machine). It also includes an introduction to measurement of the complexity of an algorithm. Part two is concerned with arithmetic complexity theory, i.e., minimizing the time required by an algorithm which computes something. It begins with a discussion of a simple problem: evaluation of a polynomial of degree N at a point with as few multiplications as possible. Evaluation of the polynomial at N points is then discussed briefly en route to a discussion of the Fast Fourier Transform, which is developed using the important technique of divide-and-conquer. This leads naturally to a discussion of recursion and recursive algorithms.

In part three we consider some non-arithmetic algorithms. Several sorting algorithms are discussed and a lower limit on the number of comparisons for sorting an arbitrary file is given.

Part four will include a summary and some speculations about the direction of research in complexity theory.

In this survey we will only consider a few problems and techniques; it is not the intention to provide a complete description of current work on complexity theory. For additional surveys see [**4**, **8**, **42**]; for a leisurely introduction to some of the ideas in algorithm design see [**20**]; and for the

Received by the editors on April 23, 1985 and in revised form on June 24, 1985.