

Introduction

An algorithm to find the greatest element of an unordered list must inspect every element of the list, or else it could not be correct. Reasoning of this kind gives us a minimum amount of work which any algorithm must do if it is to solve a problem. We are putting a lower bound on the time complexity to solve the problem. If we have already found an algorithm which does no more than this amount of work, then we can safely assert that this algorithm is optimal.

When searching through a list L or sorting it, we assume random access to L . The k^{th} element of L is denoted $L[k]$ (starting to count from 0). So we are really regarding a list as an array.