Key Terms and Results

TERMS

algorithm算法: a finite sequence of precise instructions for per- forming a computation or solving a problem
searching algorithm搜索算法: the problem of locating an element in a list
linear search algorithm线性搜索算法: a procedure for searching a list element by element
binary search algorithm二分搜索算法: a procedure for searching an ordered list by successively splitting the list in half
sorting排序: the reordering of the elements of a list into prescribed order
***f*(*x*) is *O*(*g*(*x*)):** the fact that $|f(x)| \leq C|g(x)|$ for all $x > k$ for some constants C and k
witness to the relationship见证关系 *f*(*x*) is *O*(*g*(*x*)): a pair C and k such that $|f(x)| \leq C|g(x)|$ whenever $x > k$
***f*(*x*) is $\Omega(g(x))$:** the fact that $|f(x)| \geq C|g(x)|$ for all $x > k$ for some positive constants C and k
***f*(*x*) is $\Theta(g(x))$:** the fact that $f(x)$ is both $O(g(x))$ and $\Omega(g(x))$
time complexity时间复杂度: the amount of time required for an algorithm to solve a problem
space complexity空间复杂度: the amount of space in computer memory required for an algorithm to solve a problem
worst-case time complexity最坏情况时间复杂度: the greatest amount of time required for an algorithm to solve a problem of a given size
average-case time complexity平均情况时间复杂度: the average amount of time required for an algorithm to solve a problem of a given size
algorithmic paradigm算法范式: a general approach for constructing algorithms based on a particular concept
brute force蛮力: the algorithmic paradigm based on constructing algorithms for solving problems in a naive manner from the statement of the problem and definitions

greedy algorithm贪婪算法: an algorithm that makes the best choice at each step according to some specified condition
tractable problem易解问题: a problem for which there is a worst-case polynomial-time algorithm that solves it
intractable problem难解问题: a problem for which no worst-case polynomial-time algorithm exists for solving it
solvable problem可解问题: a problem that can be solved by an algorithm
unsolvable problem不可解问题: a problem that cannot be solved by an algorithm

RESULTS

linear and binary search algorithms线性及二分查找算法: (given in Section 3.1)
bubble sort冒泡排序: a sorting that uses passes where successive items are interchanged if they in the wrong order
insertion sort插入排序: a sorting that at the jth step inserts the jth element into the correct position in in the list, when the first $j-1$ elements of the list are already sorted

The linear search has $O(n)$ worst case time complexity. The binary search has $O(\log n)$ worst case time complexity.
The bubble and insertion sorts have $O(n^2)$ worst case time complexity.
$\log n!$ is $O(n \log n)$.
If $f_1(x)$ is $O(g_1(x))$ and $f_2(x)$ is $O(g_2(x))$, then $(f_1+f_2)(x)$ is $O(\max(g_1(x), g_2(x)))$ and $(f_1 f_2)(x)$ is $O((g_1 g_2(x))$.
If $a_0, a_1, \ldots, a_n$ are real numbers with $a_n \neq 0$, then $a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ is $\Theta(x^n)$, and hence $O(x^n)$ and $\Omega(x^n)$