Algorithms for computational problems

Well-defined procedures that run in finite steps.

Well-defined input–output relationship

Sorting: input array of n integers $A = \langle a_1 \cdots a_n \rangle$

output a permutation / reordering of A $\langle a_1' \cdots a_n' \rangle$ in nondecreasing order $a_1' \le a_2' \cdots \le a_n'$

Euclidean algorithm for → GCD: input two integers a and b

output $GCD(a, b)$

Matrix multiplication: input $A = (a_{ij})$ $B = (b_{ij})$

output $C = (c_{ij}) = AB$

$$c_{ij} = \sum_{k=1}^{n} a_{ik} \cdot b_{kj}$$

Algorithms for computational problems that are "correct" and "efficient"

Correctness: provably correct on every input.

efficiency: [time as the key resource
num of basic operations needed to terminate

worst-case time complexity

RAM

Cells: numbers
pointers

basic operations:
arithmetic op.
data movement op.

Techniques for designing algorithms

[greedy
dynamic programming
randomized algorithms (hashing)
graph algorithms

data structures.

for analysis of algorithms

for giving evidence that certain problems are hard to solve.

InsertionSort : input an array $A = \langle a_1 \cdots a_n \rangle$ of n integers.

1. create an empty list B
2. For each i from 1 to n

   Go through integers in B one by one from the beginning and find the first integer larger than a_i; insert a_i right before it in the list.

constant num of steps

c_1

c_2

$c_2 k_i$

k_i : number of integers examined before finding place to insert a_i

$a_1 \cdots a_i \; a_{i+1}$

$a_1' \le \cdots \le a_i'$

in-place sorting

① correctness

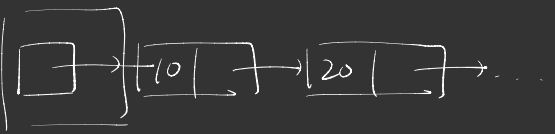
induction: At the end of each ith for loop, B contains $a_1 \cdots a_i$ and is sorted.

② efficiency

running time.

round i

$a_1' \cdots a_i' \; a_{i+1}$



Running time of Insertion sort:

worst case running time:

$$T(n) = \max_{\substack{A: \text{array of} \\ n \text{ integers}}} \left\{ \text{num of basic operations needed for InsertionSort to terminate on } A \right\}$$

worst-case

number of basic operations InsertionSort uses on A

$$c_1 + c_2 n + \sum_{i=1}^{n} c_3 k_i$$

$A = \langle 1, 2, 3, \cdots, n \rangle$ $\quad k_i = i$

B: 1, 2, 3

$$T(n) = c_1 + c_2 n + \sum_{i=1}^{n} c_3 \cdot i = c_1 + c_2 n + \frac{c_3}{2} n(n+1)$$

$$= \frac{c_3}{2} n^2 + \left(c_2 + \frac{c_3}{2}\right) n + c_1$$

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$

$A = \langle n, n-1, \cdots, 1 \rangle$

$B = \langle n-2, n-1, n \rangle$ $\quad k_i = 1$

$O(\cdot)$, $\Omega(\cdot)$, $\Theta(\cdot)$ $\quad \Theta(n^2)$