Chapter 3 - Algorithms, Integers and Matrices.

Simple computer programs, cellular automata, can explain many natural phenomena. See A new kind of science by S. Wolfram.

An algorithm is a finite set of precise instructions for performing a computation or for solving a problem.

Ex) $287 = 3 \cdot 91 + 14$

$91 = 6 \cdot 14 + 7$

$14 = 2 \cdot 7$

7 is the largest integer that divides both 287 and 91.

see 3.6

Ex) A searching problem

$a_1 = 1$, $a_2 = 3$, $a_3 = 4$, $a_4 = 5$, $a_5 = 6$, $a_6 = 8$, $a_7 = 9$, $a_8 = 11$

For which i is $a_i = 9 = x$?

Linear search: Start with a_1, if $x = a_1$ Stop otherwise continue with a_2 and so on.

If the list is orded we can do a binary search. Start with a_4. Since $x > a_4$ we compare with a_6 etc.

To investigate the time complexity of an algorithm (3.3) we have to say something about the growth of functions (3.2)

DEFINITION: $f(x)$ is big-oh of $g(x)$, or f is $O(g)$, if there are constants C and k such that

$$|f(x)| \leq C|g(x)|$$

whenever $x > k$.

3.2.6) $$\frac{x^3+2x}{2x+1} = \frac{x^2}{2} \cdot \frac{(1+2/x^2)}{(1+1/2x)} < \frac{x^2}{2}$$

when $x > 4$. So f is $O(x^2)$. We can chose $C = \frac{1}{2}$ and $k = 4$.

Ex) $n! = 1 \cdot 2 \cdot 3 \cdots n$ is $O(n^n)$ since

$$n! < \underbrace{n \cdot n \cdot n \cdots n}_{n \text{ factors}} = n^n$$

For large n (see figure 3):

$$n! > 2^n > n^2 > n \log n > n > \log n > 1.$$

Ex) $f(n) = 1 + 2 + 3 + \dots + n \leq n + n + \dots + n = n^2$.

Is $|f| > C_1 n^2$ whenever $n > k_1$ for some C_1 and k_1?

Yes, $f(n) > \frac{n}{2} + \left(\frac{n}{2}+1\right) + \dots + n = \frac{3n}{4} \cdot \left(\frac{n}{2}+1\right) > \frac{3n^2}{8}$. Same result for odd n: <u>$f(n)$ is of order n^2</u>,

3.3 Complexity of Algorithms

$10^{-9}\,s <$ time for basic bit operations $< 10^{-6}\,s$

Linear search requires at most $O(n)$ operations.

For binary search the order of comparisons is at most $\log n$.

If a problem in <u>class NP</u> can be solved by a polynomial worst-case time algorithm then all problems in this class can be solved in such a way.

10^{12} bit operations take 17 min,

2^{100} bit operations take $4 \cdot 10^3$ y,

Some interesting things in sections 3.4 - 3.8.

Pseudo-random numbers.

Ex) $x_{n+1} = 3x_n \mod 11$

Start with $x_0 = 2$.

Gives sequence

$6, 7, 10, 8, 3, 9, 5, 4, 1, 3 \ldots$

HF 3/3 2015:
Bad example.
Made a miscalculation.

Divide by modulus to get number between 0 and 1.

$x_{n+1} = 16807 x_n \mod (2^{31} - 1)$ is often used.

Binary expansions: $(10011)_2 = 1 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + 0 \cdot 2^3 + 1 \cdot 2^4 = 1 + 2 + 16 = 19$

$5 = (101)_2$

$19 \cdot 5 = 95$ or using the binary expansions

$$\begin{array}{r} 10011 \\ \times\quad 101 \\ \hline 10011 \\ 00000 \\ 10011 \\ \hline 1011111 \end{array}$$

$(1011111)_2 = 2^6 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = 64 + 31 = 95$

The time to multiply two n-digit numbers are $O(n^2)$, see also 7.3.

The chinese remainder theorem.

Ex)
$$x \equiv 2 \pmod 3$$
$$x \equiv 3 \pmod 5$$
$$x \equiv 2 \pmod 7$$

Solve first the three congurences

$x_1 \equiv 1 \pmod 3$	$x_2 \equiv 0 \pmod 3$	$x_3 \equiv 0 \pmod 3$
$x_1 \equiv 0 \pmod 5$	$x_2 \equiv 1 \pmod 5$	$x_3 \equiv 0 \pmod 5$
$x_1 \equiv 0 \pmod 7$	$x_2 \equiv 0 \pmod 7$	$x_3 \equiv 1 \pmod 7$

$x_1 = 70$ $\quad x_2 = 21$ $\quad x_3 = 15$

$$x = 2 \cdot 70 + 3 \cdot 21 + 2 \cdot 15 = 233$$

is a solution, but not the smallest one. $233 = 23 + 2 \cdot 105$. 23 is the smallest positive solution to the problem.

This representation of the integers can be used for arithmetic operations.

See Ex. 8 in 3.7. I give another example.

Ex) $23 \cdot 4 = 92$

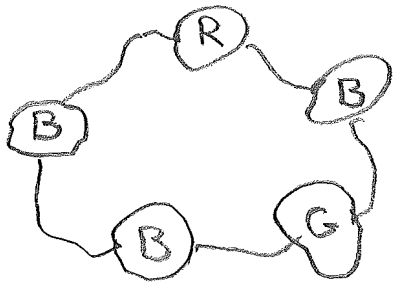
23 is represented by $(2, 3, 2)$, see above. 4 is in the same way represented by $(1, 4, 4)$. If multiply component by component we get $(2, 2, 1)$ which is the representation for 92.

Fermat's little theorem

Ex/ $3^5 \equiv 3 \pmod 5$

> $a^p \equiv a \bmod p$, p is a prime

In how many ways can a necklace with 5 bricks be colored with three colors R, B, G?



Unicolored necklaces are not allowed. The coloring in the figure, RBGBB, and the rotations BRBGB, BBRBG, GBBRB and BGBBR are regarded as the same colored necklace.

3 colors for each brick, totally 3^5 colorings. Then we take away the 3 necklaces with one color. 3^5-3 left. Then the 5 rotations should be counted as one neclace. So finally there are

$$(3^5-3)/5 = 48$$

different necklaces.

3.8.29c) $A = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, $B = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}$

ZERO-ONE MATRICES

$A \odot B = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$

↑ Boolean product

$(A \odot B)_{13} = (a_{11} \wedge b_{13}) \vee (a_{12} \wedge b_{23}) \vee (a_{13} \wedge b_{33}) = (1 \wedge 1) \vee (0 \wedge 1) \vee (1 \wedge 1) =$

$1 \vee 0 \vee 1 = 1$

see also 8.3