Algorithms

Algorithm from Algorism → 9^{th} Century Math.cian Abu Jafar Moham. Ibn Musa Al-Khawarizmi - Algebra derives also from the latin title of ~~this~~ a book written by him.

What do we mean by an algorithm? plan - sort of program - strategy

But ask yourself the question

- Does an algm. have to finish? and in a reasonable time?
- What is a reasonable ~~amoun~~ length of time
- Must it always produce the correct answer?
- " " " the same answer for the same data input
- Can we always develop an algm. to solve any given problem?

For now let's give the "definition":

"An algorithm is a finite set of instructions {operations}, each chosen from a finite set of well-defined operations, that halts in a finite time".

General characteristics: - Precision (steps precisely stated)

- Uniqueness (intermediate results of each step of execution are uniquely defined and depend only on inputs and previous step results) - Finiteness (Algm. stops after finitely many instructions have been executed) - Input - Output -
- Generality (applies a to a set of inputs).

Examples: Carrying out operation like additions - Subtractions multiplication are basic algorithms. The oldest most common and famous algorithm is Euclid's algm for calculating the greatest common divisor.

- No subjective decisions - No intuition nor creativity.

Examples: Cooking recipe is an algorithm is an algm. if precise instructions are specified "Add specific amount of..." not simply "add salt & pepper till tender".

Exception for "PROBABILISTIC ALGORITHMS"

An instruction like choose an integer between 1 and 6 is not acceptable but "choose x, $1 \leq x \leq 6$ with equal chances" is OK.

- Another exception to approximate algorithms.

Exple: $\sqrt{2}$ computed to 4 decimals.

There are pbs for which no practical algorithm is known. May be an existing one takes too long. May be obliged to look into a set a rules that we believe gives us a good approx. Some time even this is not possible and just rely on good luck. A procedure based largely on optimism and minimal theoretical support is called "heuristic Algorithm". We have no control on the error but we can estimate it.

When we solve a problem there may a choice of algorithm available and we need to decide which one to use. Depending on equipments ~~time, storage~~ and priorities choose the one with least time, minimum storage, easy to program ... and so on.

Algorithms is the science that lets us evaluate the effect of these various external factors on the available algorithms. It's also the science that tells us how to design a new algorithm for a particular task.

Exples: ① Take multiplication of 2 integers. 1) the English way vs. rest of the world way.

② Multiplication à la russe: write multiplicand and multiplicator side by side. Make 2 columns. Repeat the following rule until the number in the left hand column is 1: Divide the number in the left column by 2, ignoring any fractions, and double the number on the right column (adding it to itself). Next cross out each row where the number in the left is even, and finally add up the numbers that remain in the right column.

Exple: 12 x 28

12	28	-
6	56	-
3	112	112
1	224	224
		336

... resembles the one used in the hardware of a binary computer. All we need to know is how to add and ÷2.

(3) Divide and Conquer Multiplication.

requires same number of figure (of multiplicand & Multiplicator) add 0 to the left if needed. and number of figure should be a power of 2.

example: 981 x 1234 → 0981 x 1234

	Multiply		Shift	Result		
(i)	09	12	4	108....	left x left	shift by # of figures
(ii)	09	34	2	302..	left x right	shift by half
(iii)	81	12	2	972..	right x left	" " "
(iv)	81	34	0	2754	right x right	no shift
				1210554		

Pb. reduced to 4 multip. of 2x2. with shifts and addition.

then

	Multiply		shift	Result
(i)	0	1	2	0..
(ii)	0	2	1	0.
(iii)	9	1	1	9.
(iv)	9	2	0	18
				108

Of course does not outperform the "classic algorithm"

Notations for programs

Describe in English (plain language) not suitable. Give the corresponding program such as Pascal (or Pascal like), omit unimportant details. We'll use ÷, ⌊ ⌋. Only basic concept underlying the program. omit: declarations of scalar quantities, types of parameters and functions

Exple: function russe(m, n)

```
result ← 0
repeat
   if m is odd then result ← result + n
   m ← m ÷ 2
   n ← n + n
until m = 1
return result.
```

Mathematical notation:

1) Propositional calculus: True-false, conjunction, disjunction, negation, implication, equivalence,
2) Set Theory: sets, finite sets, cardinality, empty, Notations belongs to $\in$, Such that $|$, subset, equal, Union, intersection, difference. cartesian Product - ordered pairs, Power Set
3) Integers, Reals and intervals: $\mathbb{N} = \{0, 1, \ldots\}$, $\mathbb{Z}$, $\mathbb{R}$, $\mathbb{N}^+$, $\mathbb{R}^+$
   interval $(a, b) = \{x \mid x \in \mathbb{R}, a < x < b\}$ where $a, b \in \mathbb{R}$
   $[i..j] = \{n \mid n \in \mathbb{Z}, i \leq n \leq j\}$, $|[i..j]| = j - i + 1$
4) Relations and Functions: relation $\rho \subseteq X \times Y$, a function f
   $\forall x \in X$ $\exists$ only one $y \in Y$ s.t $(x, y) \in f$. Domain, Codomain, image Range. Injection, Surjection, Bijection. Inverse $f[f^{-1}(y)] = y$
5) Quantifiers: $\forall$, $\exists$, $\exists!$ $(\exists x \in X)[P(x)]$, $\overset{\infty}{\exists}$, $\overset{\infty}{\forall}$,
6) Sums and Products: $\sum_{i=1}^{n} f(i) = f(1) + \cdots + f(n)$, with a conditions
   $\prod_{i=1}^{n} f(i) = f(1) \times f(2) \times \cdots \times f(n)$
7) Miscellaneous: $\log_b x = y$ unique real y s.t $b^y = x$
   b can be $e = 2.7182818$, 2, 10.
   $\log(x \cdot y) = \log(x) + \log y$, $\log x^y = y \log x$
   $\log_a x = \dfrac{\log_b x}{\log_b a}$, $x^{\log y} = y^{\log x}$

   $\lfloor x \rfloor$ = integer of x $(x \geq 0)$ $\lfloor 3.5 \rfloor = 3$, $\lfloor -3.5 \rfloor = -4$ floor
   $\lceil x \rceil$ ceiling of x

- Sometimes $a \div b = \lfloor a/b \rfloor$.
- $m \bmod n = (m)_{\bmod n} = \mathrm{Rem}(m/n) = m - n.(m \div n)$
- factorial $n!$, $0! = 1$
  approximation of factorial $n! \approx \sqrt{2\pi n}\,(n/e)^n$ $e = 2.718$
- combination $\binom{m}{n}$

Proof Techniques:

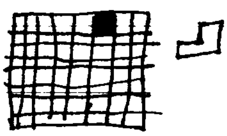
① Contradiction: "There are infinitely many Prime numbers".
② Mathematical Induction:
  Induction → Inferring of general law from particular instances.
  Deduction → Inference from general to particular.
Induction tricky a) $P(n) = n^2 + n + 41$ $P(0), P(1), \cdots, P(10) \to 41, 43, 47, 53, \ldots, 151$
all prime. but $P(40) = 1681 = 41^2$
b) $A^4 + B^4 + C^4 = D^4$ all integers Euler in 1769 conjectured no solution
in 1987 on connection Machine $95800^4 + 217519^4 + 414560^4 = 422481^4$
c) Pell's Equation given $p(n) = 991 n^2 + 1$ $\exists n$? s.t $p(n)$ is a perfect square
by trying a large number of case answer is no. But in fact
the smallest known is $n = 12055735790331359447442538767$
$(\approx 10^{29})$

Induction ~~ Experiment.
③ Principle of Mathematical induction:
Consider the following algorithm

```
function sq(n)
   if n=0 then return 0
   else return 2n + sq(n-1) -1
```

Prove by induction
$$\left(\sum_{i=1}^{n} i\right)^2 = \sum_{i=1}^{n} i^3 \quad n \geq 1$$
answer in class.

. 3 steps - Base - Assumption - Proof.
Proofs by Mathem. induction can turn into algorithms.
Consider the tiling problem: Board divided into equal squares there
are $m \times m$ squares where $m = 2^n$
one arbitrary square is special
supply of $(2\times 2 - 1)$ tiles.



Reminders:

1- LIMITS: $\lim_{n\to\infty} f(n) = a \iff \forall \delta\ (\delta \in \mathbb{R}^+)$ very small $|f(n) - a| < \delta$

. $\lim (f+g) = \lim f + \lim g$

. $\lim (f \cdot g) = \lim f \cdot \lim g$

. $\lim \frac{f}{g} = \lim f / \lim g$

De l'Hôpital Rule: Suppose $\lim_{n\to\infty} f = \lim_{n\to\infty} g = 0$ or $\lim_{n\to\infty} f = \lim_{n\to\infty} g = \infty$ and suppose that f & g are differentiable ($g' \neq 0$)

$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{f'(n)}{g'(n)}$ (after extending f&g to real fcts)

exple: $f(n) = \log n$ and $g(n) = n^a$ where $a > 0$

$\lim_{n\to\infty} f = \lim_{n\to\infty} g = \infty$ extend $f(n)$ to real function $\hat{f}(x) = \log x$

$$\lim_{n\to\infty} \frac{\log n}{n^a} = \lim_{x\to\infty} \frac{1/x}{a x^{a-1}} = \lim_{x\to\infty} \frac{1}{a x^a} = 0 \quad \forall a > 0.$$

2. Simple SERIES: $u(n)$ fct of n, $S(n) = S_n = \sum_{i=1}^{n} u_i$

$\lim_{n\to\infty} S_n = S = \sum_{i=1}^{\infty} u_i$ is convergent if $S < \infty$

- Arithmetic Series: diff^{ce} between successive terms is constant.

  $a, a+d, a+2d, \ldots, a+(n-1)d$
- Geometric Series: $a, ar, ar^2, \ldots, ar^{n-1}$ ratio btwn succ. is cst.

Sum of first n terms $S_n = an + \frac{n(n-1)d}{2}$

" " " " $S_n = a\frac{1-r^n}{1-r}$