Lecture 1: Algorithms

Algorithm derived from Algorism, 9th Century Mathematician Abu Jaafar Mohammed Ibn-Musa Al-Khawarizmi. The word Algebra derived also from the Latin title of a book written by him.

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

But ask yourself the question:
- Does an algorithm have to finish? Within a reasonable time?
- What is a reasonable length of time?
- Must it always produce the correct answer?
- Must it always produce the same answer for the same data input?
- Can we always develop an algorithm to solve any given problem?

For now let’s give the “definition”:
“An algorithm is a finite set of instruction/operations, each chosen from a finite set of well-defined operations that halts in a finite time.”

General Characteristic:
- Precision (steps precisely stated)
- Uniqueness (intermediate results of each step of execution are uniquely defined and depend only on inputs and previous step’s results)
- Finiteness (Algorithm stops after finitely many instruction have been executed)
- Input-Output

Examples: Carrying out operations like addition, subtraction and multiplication are basic algorithm. The oldest most common and famous algorithm is Euclid’s algorithm for calculating the greatest common divisor.
No subjective decisions-no intuition nor creativity.

Examples: Cooking recipe is an algorithm if precise instructions are specified. “Add specific amounts of…” not simply “add salt & pepper till tender”. The exceptions are “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.

Example: \sqrt{2} computed to 4 decimals.

There are problems for which no practical algorithm is known. Maybe an existing one takes too long. Maybe obliged to look into a set of rules that we believe gives us a good approximation. Sometimes even this is not possible and just relies on good luck. A procedure based carefully 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 is a choice of algorithms available and we need to decide which one to use. Depending on equipments and priorities choose the one with least time, minimum storage, and ease of program and so on.

Analysis of 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.

Examples:

1. Take multiplication of 2 integers, the English way and the rest of the world.
2. Multiplication ala 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 that remain is the right column.

\[
\begin{array}{ccc}
12 & 28 & - \\
6 & 56 & - \\
3 & 112 & 112 \\
1 & 224 & 224 \\
\text{total} & 336
\end{array}
\]

3. Divide and conquer Multiplication requires same number of figures (of multiplicand and multiplicator) add 0 to the left if needed and number of figures should be a power of 2.

\[
\begin{array}{cccc}
981 & \times & 1234 & \rightarrow 0981 \times 1234 \\
\text{Multiply} & \text{Shift} & \text{Result}
\end{array}
\]

\[
\begin{array}{cccc}
i. & 09 & 12 & 4 & 108… \text{left x left} & \text{shift by # of figures} \\
ii. & 09 & 34 & 2 & 302…\text{left x right} & \text{shift by half} \\
iii. & 81 & 12 & 2 & 972…\text{right x left} & \text{shift by half} \\
iv. & 81 & 34 & 0 & 2754…\text{right x right} & \text{no shift} \\
\text{total} & & & 1210554
\end{array}
\]

Problem reduced to 4 multiplications of 2 x 2 with shift and addition, then

\[
\begin{array}{cccc}
\text{Multiply} & \text{Shift} & \text{Result}
\end{array}
\]

\[
\begin{array}{cccc}
i. & 0 & 1 & 2 & 0 \\
ii. & 0 & 2 & 1 & 0 \\
iii. & 9 & 1 & 1 & 9 \\
iv. & 9 & 2 & 0 & 18 \\
\text{total} & & & 108
\end{array}
\]

Of course does not out perform the “classic algorithm”

**Notations for programs**
Describe in English (plain language). Give the corresponsive program such as Pascal (or Pascal like), omit unimportant details we’ll use \( \div, \lceil, \rceil \). Only have concept underlying the program. Omit declarations of scalar quantities, types of parameters and functions.

Example: function russe (m,n)

Result \( \leftarrow 0 \)

Repeat

If m is odd then result \( \leftarrow \) result + n

\( m \leftarrow m \div 2 \)

\( n \leftarrow n + n \)

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 pair, power set.
3. Integers, real and intervals: \( \mathbb{R} = \{0,1\ldots \} \), \( \mathbb{Z} \), \( \mathbb{R}^+, \mathbb{S}^+ \)
   Interval \( (a,b) = \{x|x \in \mathbb{R}, a<x<b\} \) where \( a,b \in \mathbb{R} \)
   Interval \( [i..j] = \{n|n \in \mathbb{Z}, i \leq n \leq j\} \), \( |[i..j]| = j-i+1 \)
4. Relations and Functions: relation \( f \subseteq \mathbb{X} \times \mathbb{Y} \), a function \( f \forall x \in \mathbb{X} \exists y \in \mathbb{Y} \) s.t \( (x,y) \in f \). Domain, co-domain, imagine, range, injection, surjection, projection. Inverse \( f^{-1}(y) = y \).
5. Quantifiers: \( \forall, \exists, \exists! (\exists x \in \mathbb{X} [P(x)] \), \( \infty, \forall \)
6. Sums and Products: \( \sum_{i=1}^{n} f(i) + \ldots + f(n) \) with a condition \( \prod_{i=1}^{n} f(i) \cdot f(2) \ldots f(n) \)
7. Miscellaneous: \( \log_b x = y \) unique real y s.t \( b^y = x \) b can be \( e = 2.7182818, 2, 10 \)
   - \( \log(x \cdot y) = \log(x) + \log(y) \)
   - \( \log(x)^y = y \cdot \log(x) \)
   - \( \log_a (x) = \frac{\log_b (x)}{\log_b (a)} \)
   - \( (x)^{\log(y)} = (y)^{\log(x)} \)
   - \( \lceil x \rceil = \) integer of \( x \) \( (x \geq 0) \) \( \lceil 3.5 \rceil = 3, \lceil -3.5 \rceil = 4 \) floor
   - \( \lfloor x \rfloor = \) ceiling of \( x \)
   - sometimes \( \lfloor a \div b \lfloor = [a/b] \)
   - \( m \) and \( n = (m) \mod n = \) Run \( (m/n) = m-n \cdot (m\div n) \)
   - factorial \( n! \), \( 0! = 1 \) approximation of factorial \( n! \approx 2\pi (n/e) \) \( e = 2.718 \)
   - combination \( (m \binom{n}{} \)

Proof Techniques:
1. Contradiction: “There are infinitely many prime numbers”.

2. Mathematical Induction:
   - Induction $\rightarrow$ inferring of general laws from particular instances.
   - Deduction $\rightarrow$ inferring from general to particular.

   Induction trick:
   a) $P(n) = n^2 + n + 41$ for $P(0), P(1), \ldots P(10)$
      
      $44, 43, 47, 53\ldots 151$ 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 $P(n) = 991n^2 + 1 \exists n? P(n)$ is a perfect square by trying a larger number of cases, the answer is no. In fact the smallest known answer is $n=12055735790631359447442538767 \approx 10^{29}$

3. Principle of Mathematical induction:
   - consider the following algorithm
   - function sq(n)
     - if $n=0$ then return 0
     - else return $2n + \text{sq}(n-1) - 1$
   
   Prove by induction: $(\sum_{i=1}^{n} i)^2 = \sum_{i=3}^{n} i \geq 1$ (answer in class)

3 steps – Base – Assumption – Proof

Proofs by mathematical 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: $\lim_{n \to \infty} f(n) = a \iff \forall \delta (\delta \in \mathbb{R}^+) \text{ very small } |f(n) - a| < \delta$
   
   $\lim(f+g) = \lim f + \lim g$
   $\lim(f \cdot g) = \lim f \cdot \lim g$
   $\lim(f/g) = \lim f / \lim g$
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 \( f \) and \( g \) are differentiable \((g \neq n)\) \( \lim f(n)/g(n) = \lim f'(n)/g'(n) \) (after extending \( f \) & \( g \) to real functions).

Example: \( f(n) = \log n \) and \( g(n) = n^a \) where \( a > 0 \). Extend \( f(n) \) to real function \( \log(n^a) = \lim_{n \to \infty} \frac{\log(n)}{n^a} = \lim_{n \to \infty} \frac{x}{ax^{a-1}} = 0 \) \( \forall a > 0 \).

2. Simple Series: \( u(n) \) function of \( n \), \( s(n) = s_n = \sum_{i=1}^{n} u_i \), \( \lim s_n = s = \sum_{i=1}^{\infty} u_i \) is convergent if \( s < \infty \).
   a. Arithmetic series: difference between successive term is constant \( a, a+d, a+2d, \ldots a+(n-1)d \)
   b. Geometric series: \( a, ar, ar^2, \ldots, ar^{n-1} \) ratio between successive is constant. The sum of first \( a \) terms \( s_n = an + \frac{n(n-1)d}{2} \),

   \[
   \text{sum of first a term } s_n = a + \frac{1-r^n}{1-r}
   \]