CSC006P1M: Design and Analysis of Algorithms
Lecture 01 *(Introduction to Algorithms)*

Sumit Kumar Pandey

August 08, 2022

What is Algorithm?

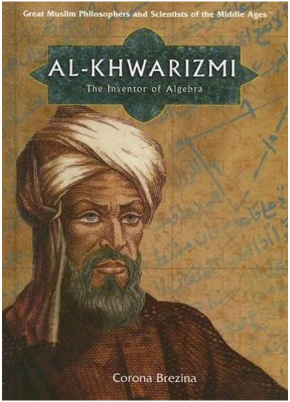
Definition

An algorithm is a finite set of instructions that if followed, accomplishes a particular task. In addition, all algorithms must satisfy the following criteria:

1. **Input** - Zero or more quantities are externally supplied.
2. **Output** - At least one quantity is produced.
3. **Definiteness** - Each instruction is clear and unambiguous.
4. **Finiteness** - If we trace out the instructions of an algorithm, then for all input cases, the algorithm terminates after a finite number of steps.
5. **Effectiveness** - Every instruction must be very basic so that it can be carried out, in principle, by using only a pencil and a paper. It is not enough that each operation be definite as in criterion 3; it also must be feasible.

History

The word algorithm comes from the name of a Persian author, Abu Ja'far Mohammed ibn Musa al Khowarizmi (825 A.D.), who wrote a textbook on mathematics.



Evaluating Polynomials

Problem

Given a sequence of real numbers $a_n, a_{n-1}, \cdots, a_1, a_0$, and a real number x, compute the value of the polynomial
$P_n(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$.

Example: Let $x = 3$, $n = 5$, $a_0 = 1$, $a_1 = 2$, $a_2 = 10$, $a_3 = 4$, $a_4 = 9$ and $a_5 = 2$
Then, $P_5(3) = 2 * 3^5 + 9 * 3^4 + 4 * 3^3 + 10 * 3^2 + 2 * 3^1 + 1$
How to compute $P_5(3)$?

Solution

Let,

- $T_0 = 1$
- $T_1 = 2 * 3$
- $T_2 = 10 * 3 * 3$
- $T_3 = 4 * 3 * 3 * 3$
- $T_4 = 9 * 3 * 3 * 3 * 3$
- $T_5 = 2 * 3 * 3 * 3 * 3 * 3$
- $P_5(3) = T_5 + T_4 + T_3 + T_2 + T_1 + T_0$

So, the total number of

- Multiplications $= 1 + 2 + 3 + 4 + 5 = 15$
- Additions $= 5$

Evaluating Polynomials

Looking through the mathematical induction.
$P_n(x) = P_{n-1}(x) + a_n * x^n$

Induction Hypothesis

We know how to compute $P_{n-1}(x)$.

Then, the total number of

- Multiplications = Total number of multiplications in $P_{n-1}(x) + n$
- Addition = Total number of additions in $P_{n-1}(x) + 1$

Evaluating Polynomials

In general, for $P_n(x)$, the total number of

- Multiplications $= 1 + 2 + \cdots + n = \frac{n(n+1)}{2}$
- Additions $= n$

Can we do better?

Solution

Let,

- $T_0 = 1$
- $S_1 = 3$ and $T_1 = 2 * S_1$
- $S_2 = S_1 * 3$ and $T_2 = 10 * S_2$
- $S_3 = S_2 * 3$ and $T_3 = 4 * S_3$
- $S_4 = S_3 * 3$ and $T_4 = 9 * S_4$
- $S_5 = S_4 * 3$ and $T_5 = 2 * S_5$
- $P_5(3) = T_5 + T_4 + T_3 + T_2 + T_1 + T_0$

So, the total number of

- Multiplications $= 4 + 5 = 9$
- Additions $= 5$

Evaluating Polynomials

Looking through the mathematical induction.
$P_n(x) = P_{n-1}(x) + a_n * x^{n-1} * x$

Stronger Induction Hypothesis

We know how to compute $P_{n-1}(x)$ and x^{n-1}.

Then, the total number of

- Multiplications = Total number of multiplications in $P_{n-1}(x) + 1 + 1$
- Addition = Total number of additions in $P_{n-1}(x) + 1$

In general, for $P_n(x)$, the total number of

- Multiplications $= n - 1 + n = 2n - 1$
- Additions $= n$

Can we do better?

Solution

Let,

- $T_4 = 2 * 3 + 9$
- $T_3 = T_4 * 3 + 4$
- $T_2 = T_3 * 3 + 10$
- $T_1 = T_2 * 3 + 2$
- $T_0 = T_1 * 3 + 1$
- $P_5(3) = T_0$

So, the total number of

- Multiplications $= 5$
- Additions $= 5$

Evaluating Polynomials

Looking through the mathematical induction.
$P_n(x) = (P'_{n-1}(x)) * x + a_0$, where
$P'_{n-1}(x) = a_n x^{n-1} + a_{n-1} x^{n-2} + \cdots + a_1$

Induction Hypothesis (Reverse Order)

We know how to compute $P'_{n-1}(x)$.

Then, the total number of

- Multiplications = Total number of multiplications in $P'_{n-1}(x) + 1$
- Addition = Total number of additions in $P'_{n-1}(x) + 1$

In general, for $P_n(x)$, the total number of

- Multiplications $= n$
- Additions $= n$

This algorithm for evaluating polynomial is known as **Horner's rule** after the English mathematician W.G. Horner.

Algorithm(Horner's rule)

- Input: $a_0, a_1, a_2, \cdots, a_n$ and x.
- Output: P

begin
$P := a_n$;
for $i := 1$ to n do
$P := x * P + a_{n-i}$
end

Finding One to One Mappings

Let f be a map that maps a finite set A into itself (i.e. every element of A is mapped to another element of A). For simplicity, we denote the elements of A by the integers 1 to n.

Problem

Given a finite set A and a mapping from A to itself, find a subset $S \subseteq A$ with the maximum number of elements, such that (1) the function f maps every element of S to another element of S (i.e., f maps S into itself), and (2) no two elements of S are mapped to the same element (i.e., f is one-to-one when restricted to S).

Finding One to One Mappings

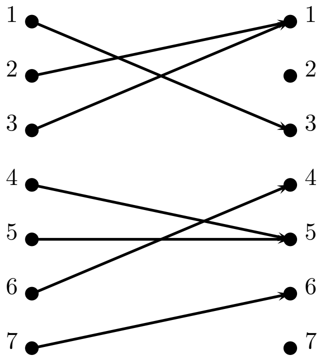


Figure: A mapping from a set into itself

The solution for above figure is $S = \{1, 3, 5\}$.

Solution

If f is originally one to one, then $S = A$, otherwise

Use mathematical induction to solve the problem.

Induction Hypothesis

We know how to solve the problem for sets of $n - 1$ elements.

Solution

The base case, i.e. $n = 1$, is trivial; if there is only one element in the set, then it must be mapped to itself, which is a one-to-one mapping.

We assume that A is not one-one. We claim that any element i that has no other element mapped to it cannot belong to S. Otherwise, if $i \in S$ and S has, say, k elements, then those k elements are mapped into at most $k - 1$ elements; therefore the mapping cannot be one-one. If there is any such i, then we simply remove from the set. Let $A' = A - \{i\}$ with $n - 1$ elements, which f maps into itself.

By induction hypothesis, we know how to solve the problem for A'. If no such i exists, then the mapping is one-one, and we are done.

Thank You