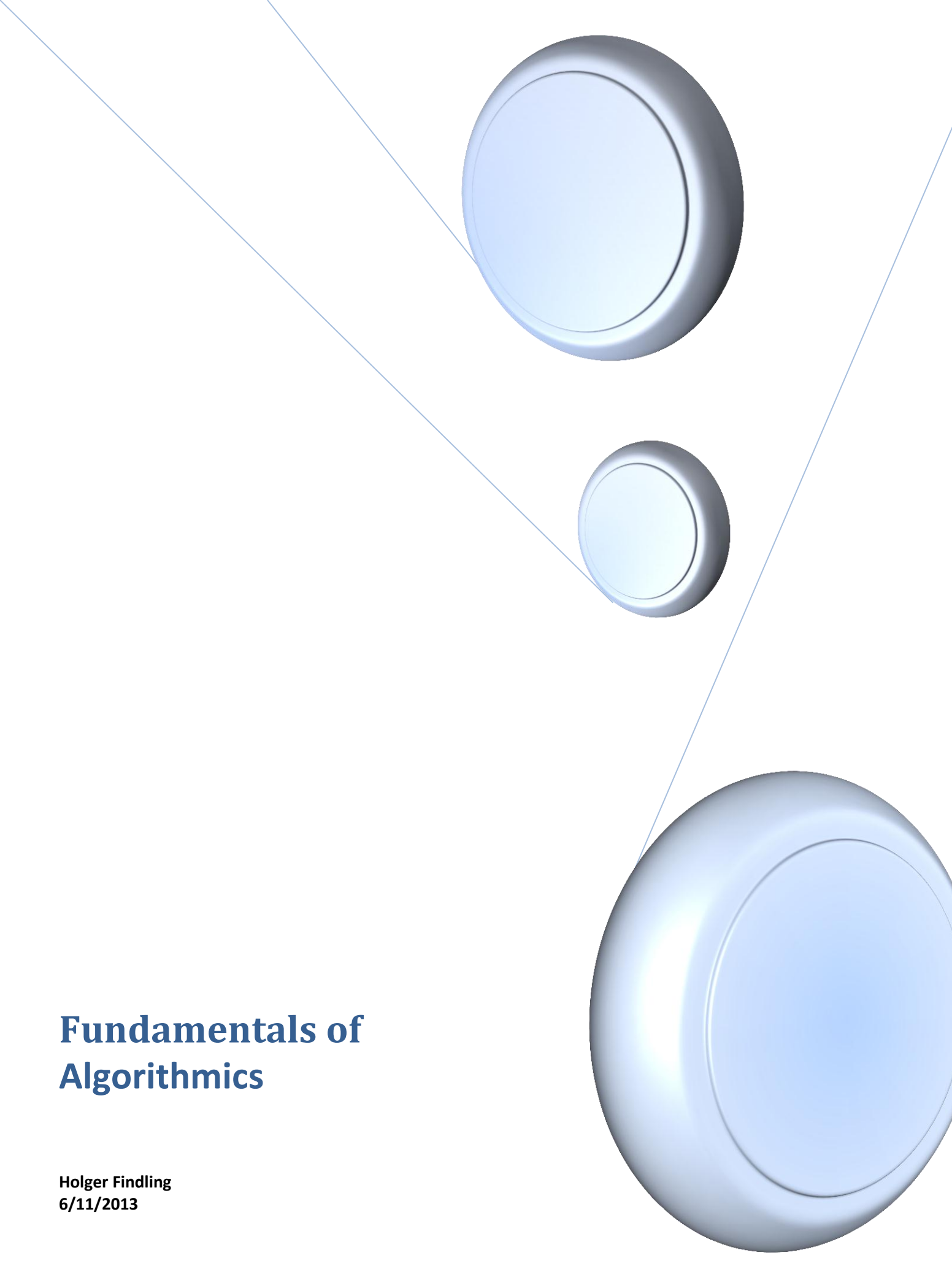

Fundamentals of Algorithmics

Holger Findling
6/11/2013

Contents

Introduction

The word algorithm was entered in the Webster's New World Dictionary about 1957. Prior to 1957 the word algorism was used to describe the process of doing arithmetic using Arabic numerals.

Historians of mathematics believe that they found the true origin of the word algorism. The title of a Persian textbook "Kitab al jabr wa'l-muqabala" (Rules of restoring and equating) may be the originator of the word "algebra". The book provides a systematic study of the solution of linear and quadratic equations.

The author's name is Abu 'Abd Allah Muhammmed ibn Musa al-Khwarizm, or translated literally, "Father of Abdullah, Mohammad, son of Moses, native of Khwarizm (c. 825).

Definition of Algorithms

The modern meaning of algorithm is similar to that of recipe, process, method, technique, or rigmarole.

Besides merely being a finite set of rules that gives a sequence of operations for solving a specific type of problem, an algorithm has five important features:

1) **Finiteness** – An algorithm must always terminate after a finite number of steps. A procedure that lacks finiteness may be called a computational method.

2) **Definiteness** - Each step of an algorithm must be precisely defined; the actions to be carried out must be rigorously and unambiguously specified for each case.

3) **Input** – An algorithm has zero or more inputs: quantities that are given to it initially before the algorithm begins, or dynamically as the algorithm runs. These inputs are taken form specified sets of objects.

4) **Output** – A algorithm has one or more outputs: quantities that have a specified relation to the inputs.

5) **Effectiveness** – An algorithm is also generally expected to be effective, in the sense that its operations must all be sufficiently basic that they can in principal be done exactly and in a finite length of time by someone using pencil and paper.

Formal Models

Analysis of Algorithms is the method used to determine performance characteristics of an algorithm or a program. The general idea is to take a particular algorithm and to determine its quantitative behavior; occasionally we also study whether or not an algorithm is optimal in some sense.

There are various formal models of what constitutes an algorithm:

- Turing Machines are the most famous. They are abstract models for a machine capable of computing any "computable" function.
- The Church-Turing thesis assures us that a problem is solvable only if there is a suitably designed Turing machine that solves it. To show that a problem has no solution **reduces** to demonstrating that no Turing machine can be designed to solve the problem.
- Random Access Machines (RAM) correspond best to our concept of a computer. RAM are also equivalent in computing power to Turing machines.
- λ- Calculus is the formalism behind some programming languages

Computability

The investigation of computability was motivated by two fundamental questions:

1. What is an algorithm?
2. What are the capabilities and limitations of algorithmic computations?

Once a problem is known to be solvable, one can begin to consider the efficiency or optimality of a solution. Time complexity is used to partition the set of solvable problems into two classes:

1. Tractable
2. Intractable.

A problem is considered tractable if it is solvable by a Turing machine in which the number of instructions executed during a computation is bounded by a polynomial function of length of the input.

A problem that is not solvable in polynomial time is considered intractable because of the excessive amount of computational resources required to solve all but the simplest cases of the problem.

Static View

A Static View of Algorithmic Problem Solving

We can identify a few major aspects of algorithmic problem solving.

They are:

- Computational means – the principal choices of interest here are sequential vs. parallel computers; more exotic options include a human and such abstract devices as Turing machines.
- Exact vs. approximate solving.
    - Example, nonlinear equations cannot be solved exactly by any algorithm.
- Deterministic vs. probabilistic paradigm.
- Data structures.
- Algorithm design techniques.

There are several dimensions that are applicable to an algorithm in the analysis stage, as opposed to the design stage.

- Correctness (and a degree of error for an approximation algorithm.)
- Efficiency (time and space.)
- Clarity (simplicity.)
- Optimality

A Simple Analysis Example

```
double  distance (double x, double y)
{
        double Result = 0.0;                    1 line of code
        double Xsquared = x * x;                1 line of code
        double Ysquared = y * y;                1 line of code
        Result = Xsquared + Ysquared;           1 line of code
        Result = sqrt (Result);                 5 NOP + Time (sqrt())
        return Result;                          1 line of code
}
```

Assume that the function sqrt() is loaded into CPU Registers in 5 Clock Cycles and executes in 19 clock cycles.

Sqrt() = 24 Clock cycles

Time () = (5 LOC * 19 Clock Cycles) + 24 Clock Cycles

Time () = 119 Clock Cycles

Assume the Computer Clock = 1 GHZ

Time () = 119 Clock Cycles * 1 E -9 seconds

Time () = 119 nanoseconds

Note: most emulators provide a very accurate count of clock cycles required to run code.

See TI DSP Processors – TMS320C6400.

Loops

```
double Sum (int *array, int size)
{
        int i                                   declaration ( 0 LOC )
        int *ptr = array;                       1 LOC
        int sum = 0;                            1 LOC

                                                size - 1
        for (i = 0; i < size; i++)              ∑ 4
                                                i = 0
        {
            sum += *ptr;                        1 LOC
            ptr++;                              1 LOC
        }

        return sum;                             1 LOC
}
```

Assume size = n

$$T(n) = 2 + \sum_{i=0}^{n-1} 4 + 1$$

$$T(n) = \sum_{i=1}^{n} 4 + 3$$

$T(n) = 4n + 3$

Review – Sigma Notation

The sum $1^2 + 2^2 + 3^2 + 4^2 + 5^2$ can be expressed by

$$\sum_{k=1}^{5} k^2 = 1^2 + 2^2 + 3^2 + 4^2 + 5^2$$

If the upper and lower limits of a summation are the same, then "sum" reduces to one term.

$$\sum_{k=2}^{2} k^3 = 2^3$$

If the expression to the right of $\sum$ does not involve the index of summation, then we take all the terms in the sum to be the same.

$$\sum_{k=1}^{5} 2 = 2 + 2 + 2 + 2 + 2$$

$$\sum_{k=3}^{6} 7 = 7 + 7 + 7 + 7$$

$$\sum_{j=0}^{2} x^3 = x^3 + x^3 + x^3$$

A sum can be written in various ways with the sigma notation, depending on the limits of summation.

$$\sum_{k=1}^{5} 2k = 2 + 4 + 6 + 8 + 10$$

$$\sum_{k=0}^{4} (2k + 2) = 2 + 4 + 6 + 8 + 10$$

$$\sum_{k=2}^{6} (2k - 2) = 2 + 4 + 6 + 8 + 10$$

Sometimes we need to change the sigma notation for a given sum to a sigma notation with different summation limits.

Example:

Express the sigma notation where the lower limit is 0 rather than 3.

$$\sum_{k=3}^{7} 5k - 2 = 13 + 18 + 23 + 28 + 33 = 115$$

Let $j = k - 3$, then $k = j + 3$

$$\sum_{j=0}^{4} 5(j + 3) - 2 = \sum_{j=0}^{4} 5j + 13$$

$$\sum_{j=0}^{4} 5j + 13 = 15 + 18 + 23 + 28 + 33 = 115$$

We can change the sigma notation back to read

$$\sum_{k=0}^{4} 5k + 13$$

Theorems for manipulating sigma notation

$$\sum_{k=1}^{n} (ak + bk) = \sum_{k=1}^{n} ak + \sum_{k=1}^{n} bk$$

$$\sum_{k=1}^{n} (ak - bk) = \sum_{k=1}^{n} ak - \sum_{k=1}^{n} bk$$

$$\sum_{k=1}^{n} cak = c \sum_{k=1}^{n} ak$$

$$\sum_{k=1}^{n} ak \sum_{k=1}^{n} bk = \sum_{k=1}^{n} \left(\sum_{k=1}^{n} ak\, bk \right)$$

Common Summations

$$\sum_{k=1}^{n} 1 = 1 + 1 + 1 + \ldots + 1 = n * 1 = n$$

$$\sum_{k=1}^{n} k = 1 + 2 + 3 + \ldots + n = \tfrac{1}{2} n (n + 1)$$

$$\sum_{k=1}^{n} k^2 = 1^2 + 2^2 + 3^2 + \ldots + n^2 = 1/6\, n (n + 1) (2n + 1)$$

$$\sum_{k=1}^{n} k^3 = 1^3 + 2^3 + 3^3 + \ldots + n^3 = [\ \tfrac{1}{2} n (n + 1)\ ]^2$$

$$\sum_{k=1}^{n+p} k = \tfrac{1}{2} (n + p) (n + p + 1)$$

Problems

Example 1

Solve for S (n)

$$S(n) = \sum_{k=1}^{n} (k + 2)$$

$$S(n) = \sum_{k=1}^{n} k + \sum_{k=1}^{n} 2$$

$$S(n) = \tfrac{1}{2} [n (n + 1)] + 2n$$

$$S(n) = \tfrac{1}{2} n (n + 5)$$

Example 2

Solve for S (n)

$$S(n) = \sum_{k=1}^{n} (12k + 2)$$

$$S(n) = 12 \left(\sum_{k=1}^{n} k + \sum_{k=1}^{n} 2 \right)$$

$$S(n) = 12 \sum_{k=1}^{n} k + 2n$$

$$S(n) = 12 [\tfrac{1}{2} n(n + 1)] + 2n$$

$$S(n) = 6 n^2 + 8 n$$

$$S(n) = 2 n (3 n + 4)$$

Arithmetic Series

$$\sum_{i=1}^{n} (a + id) = (a + 1d) + (a + 2d) + (a + 3d) + \ldots + (a + [n-1] d) + \ldots = an + \tfrac{1}{2} d n (n + 1)$$

Proof:

$$\sum_{i=1}^{n} a + d \sum_{i=1}^{n} i = an + \tfrac{1}{2} d n(n + 1)$$

$$S_n = an + \tfrac{1}{2} d n (n + 1)$$

Geometric Series

$a + ar + ar^2 + \ldots + ar^{k-1} + \ldots$ where a != 0

Converges if $|r| < 1$ and diverges if $|r| \geq 1$.
When the series converges the sum is

$a / (1 - r) = a + ar^2 + \ldots + ar^{k-1} + \ldots$

Proof:

- Let us treat the case $|r| = 1$. If r = 1, then the series is $a + a + a + a + \ldots + a + \ldots$

So that the n*th* partial sum is $S_n = na$.

$\lim_{n \to +\infty} S_n = \lim_{n \to +\infty} (na) = \pm\infty$

This proves divergence.

- Let us treat the case r = -1.

If r = -1, then the series is $a - a + a - a + \ldots$

So the sequence of partial sums is a, 0, a, 0, a, … which diverges.

- Let us consider the case where $|r| \neq 1$. T

The nth partial sum of the series is
$S_n = a + ar + ar^2 + \ldots + ar^{n-1}$ --- #1

Multiply both sides of the equation by r, it yields
$rS_n = ar + ar^2 + \ldots + ar^{n-1} + ar^n$ --- #2

Subtracting #2 from #1 gives
$S_n - rS_n = a - ar^n$ or

$(1 - r) S_n = a - ar^n$

Since $r \neq 1$in the case we are considering, the equation can be rewritten as

$S_n = (1/1-r) (a - a r^n) = (1/1-r) (a) - (1/1-r) (a r^n)$

If $|r| < 1$, then $\lim_{n \to +\infty} r^n = 0$

$\lim_{n \to +\infty} S_n = (1/1-r) a$

$$\sum_{i=0}^{n} r^i = (1/1-r)\, a$$

Harmonic Series

One of the most famous and important diverging series is the harmonic series.

$$\sum_{k=1}^{\infty} 1/k = 1 + \tfrac{1}{2} + 1/3 + \tfrac{1}{4} + 1/5 + 1/6 + \ldots$$

Because the terms in the series are all positive, the partial sums are

$S_1 = 1$
$S_2 = 1 + \tfrac{1}{2}$
$S_3 = 1 + \tfrac{1}{2} + 1/3$

The partial sums form an increasing sequence
$S_1 < S_2 < S_3 < \ldots < S_n$

The name harmonic is derived from music: the k^{th} harmonic of a string is the tone produced by a string that is 1/k times as long as the first string.

An asymptotic formula for Harmonic numbers:

$S_n = \ln(n) + \gamma + 1/(2n) - 1/(12 n^2) + 1/(120 n^4) + \ldots = O(\lg(n))$

Where $\gamma = 0.577215664901\ldots$ is Euler's constant.

Logarithmic Functions

We need to be aware of the following identities.

Please note:

I will often refer to **log,** by which I imply base 10.

Also when I write **lg**, I mean to imply $\log_2$.

$\log_b (x\, y) = \log_b x + \log_b y$

$\log_b (x/ y) = \log_b x - \log_b y$

$\log_b x^n = n \log_b x$

$\log_x b = 1/ \log_b x$

$\log_b x = \log_b a \log_a x$

$x^{\lg y} = y^{\lg x}$