Lecture 2 – Maths for Computer Science Recurrences

Denis TRYSTRAM
Lecture notes MoSIG1

sept. 19, 2019

Objective

The objective of this lecture is to present induction, which is the basic principle of recurrence and to show its application on several examples.

This lecture is devoted to deriving and solving a variety of types of recurrences (linear recurrence, multiple steps recurrences, etc.).

We will show how to use various ways for solving.

Brief overview of this sequence

- Recall of the induction principle.
- A first example (direct application)
- Take care of the first steps!
- Sum of cubes
- Master Theorem
- Non linear recurrences:
  Hanoi's towers and The token game

An example of induction: Factorial

The classical first example of the *recurrent* mode of computing involves the *factorial function* $\textsc{Fact}$ (of a nonnegative integer). The “direct” mode of computing $\textsc{Fact}$ at an argument n is:

$$\textsc{Fact}(n) \ = \ 1 \times 2 \times \cdots \times n.$$

The *recurrent* mode of computing $\textsc{Fact}(n)$ is more compact—and it better exposes the inherent structure of the function.

$$\textsc{Fact}(n) \ = \ \begin{cases} n \times \textsc{Fact}(n-1) & \text{if} \ \ n > 1 \\ 1 & \text{if} \ \ n = 1 \end{cases}$$

Proof by recurrence

Goal: proving that a statement *P(n)* involving integer n is true using the induction principle.

- **Basis.** Solve the statement for the small values of n.
- **Induction step.** Prove the statement for n assuming it is correct for $k \leq n - 1$.

Proposition. $\forall n$, the nth perfect square is the sum of the first n odd integers.

$$n^2 \;=\; 1 \;+\; 3 \;+\; 5 \;+\cdots+\; (2n-3) \;+\; (2n-1)$$

Proof. For every positive integer m, let $\mathbf{P}(m)$ denote the assertion

$$m^2 \;=\; 1+3+5+\cdots+(2m-1).$$

Let us proceed according to the standard format of an inductive argument.

- **Basis.** Because $1^2 = 1$, proposition $\mathbf{P}(1)$ is true.
- **Induction step.** Let us assume, for the sake of induction, that assertion $\mathbf{P}(m)$ is true for all positive integers strictly smaller than n.

Consider now the summation

$$1+3+5+\cdots+(2n-3)+(2n-1)$$

Because $\mathbf{P}(n-1)$ is true, we know that

$$\begin{aligned}1+3+\cdots+(2n-1) &= \big(1+3+\cdots+(2n-3)\big)+(2n-1)\\ &= \big(1+3+\cdots+(2(n-1)-1)\big)+(2n-1)\\ &= (n-1)^2+(2n-1)\end{aligned}$$

By –easy– direct calculation, we now find that

$$(n-1)^2+(2n-1) \;=\; (n^2-2n+1)+(2n-1) \;=\; n^2$$

The Principle of (Finite) Induction tells us that $\mathbf{P}(n)$ is true for all integer n.

Starting an induction

Let us (mis)use the method of Finite Induction to craft a fallacious proof of the following absurd “fact”.

Proposition. All horses are the same color.

The base case. If there is only a single horse in the set, then P is true (*all* horses in the set are the same color).

Inductive hypothesis. in every set of n horses, all horses in the set are the same color.

Extension to $n + 1$ horses.

Let us be given a set of $n + 1$ horses.
Remove one horse from the set, then the remaining set, call it S has n horses.
By our inductive hypothesis, all of the horses in S have the same color.
Now remove one horse from S and replace it with the horse that was removed from the $(n + 1)$-horse set. We now have a new n-horses set S'.
Once again we invoke the inductive hypothesis to conclude that all horses in S' have the same color. If we now reunite all of the horses, the *transitivity* of the relation "have the same color" guarantees that all of the horses in the $(n + 1)$-horses set have the same color.

What's wrong?

We all know that all horses do *not* share the same color.

What's wrong?

We all know that all horses do *not* share the same color.
The base case was not adequate for the "proof"

When we remove first one horse from the set of $n + 1$ horses and then another horse from that set, and we still have a horse left to compare those two horses to, *we must have started with at least* **three** *horses!*
This means that $n + 1$ *must be no smaller than* 3*, so* n *must be no smaller than* 2*.* The base of the induction must, therefore, be sets that contain 2 horses—and the same-color "proposition" is, of course, absurd for such sets!

This brings us to the critical issue of how to select the "small" cases that comprise the base of our induction.

Sum of n first cubes

$C_n = \sum_{k=1}^{n} k^3$

A straightforward upper bound is n^4 since each of the n terms of the sum k^3 is less than n^3.
This bound may be refined using the analogy of integrals and Riemann's sum, which leads to a value of order $\frac{n^4}{4}$.

Let us determine precisely this expression.

The first ranks give us an idea:
$C_1 = 1$
$C_2 = 1 + 8 = 9 = 3^2$
$C_3 = 1 + 8 + 27 = 36 = 6^2$
$C_4 = 1 + 8 + 27 + 64 = 100 = 10^2$, ...

The first ranks give us an idea:
$C_1 = 1$
$C_2 = 1 + 8 = 9 = 3^2$
$C_3 = 1 + 8 + 27 = 36 = 6^2$
$C_4 = 1 + 8 + 27 + 64 = 100 = 10^2$, ...

All these values are perfect squares.

The first ranks give us an idea:
$C_1 = 1$
$C_2 = 1 + 8 = 9 = 3^2$
$C_3 = 1 + 8 + 27 = 36 = 6^2$
$C_4 = 1 + 8 + 27 + 64 = 100 = 10^2$, ...

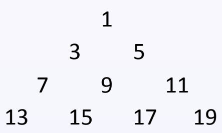
All these values are perfect squares.
A more attentive observation evidences a link with the triangular numbers (1,3,6,10, ...):
$C_n = \Delta_n^2$

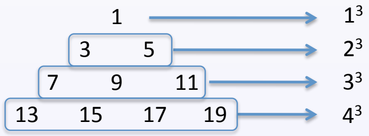
This is a guess, **not a proof!**.
We can do a simple recurrence...

Link with summing the odd numbers

We proved that summing the odd numbers is equal to a perfect square.

The figure below shows an organization of the odd numbers (left) where each row corresponds to a cube of the number of elements in the row (right).

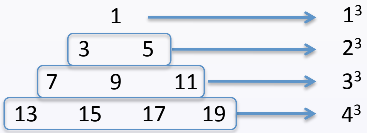




There are k numbers in row k.

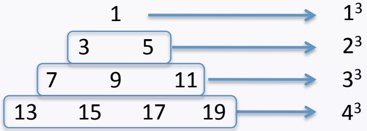
A little more effort...

The summation over each column, which makes the cubes is not obvious.



A little more effort...

The summation over each column, which makes the cubes is not obvious.

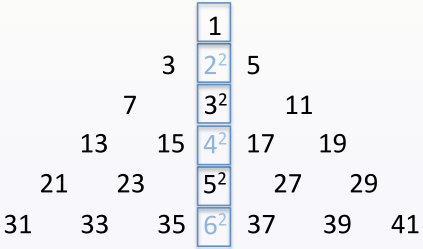


Row k contains the sum of odds from $\Delta_{k-1} + 1$ to Δ_k.
$= (\frac{k(k+1)}{2})^2 - (\frac{k(k-1)}{2})^2 = \frac{k^2}{4}((k+1)^2 - (k-1)^2) = \frac{k^2}{4}(4k) = k^3$

A structural evidence?

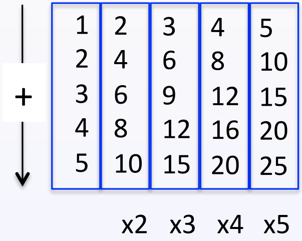
Let us look at row *k* more carefully.

$$\begin{array}{c}
1 \\
3 \qquad 5 \\
7 \qquad 9 \qquad 11 \\
13 \qquad 15 \qquad 17 \qquad 19 \\
21 \qquad 23 \qquad 25 \qquad 27 \qquad 29 \\
31 \qquad 33 \qquad 35 \qquad 37 \qquad 39 \qquad 41
\end{array}$$


1
3 2^2 5
7 3^2 11
13 15 4^2 17 19
21 23 5^2 27 29
31 33 35 6^2 37 39 41

Using the double counting principle

An alternative proof is to determine this result directly by using multiplicative tables (each row/column is multiplied by the next integer).

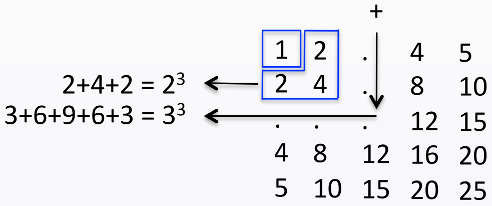


1	2	3	4	5
2	4	6	8	10
3	6	9	12	15
4	8	12	16	20
5	10	15	20	25
	x2	x3	x4	x5

Using classical multiplication tables

The sum of all the elements in column k of this matrix is equal to k times the triangular number Δ_n. Thus, the global sum is:
$1.\Delta_n + 2.\Delta_n + ... + n.\Delta_n = (1 + 2 + ... + n).\Delta_n = \Delta_n^2$.

Using Fubini double counting principle, it is easy to remark that the same global sum can also be obtained by summing the n quadrants:



Each of these partial sums at rank k is k times the ascending/descending triangular numbers: $k.(1+2+...+(k-1)+k+(k-1)+...+2+1) = k.(\Delta_k + \Delta_{k-1})$. From the previous result $\Delta_k + \Delta_{k-1} = k^2$, the sum of the elements in quadrant k is k^3. Thus, the global sum is equal to $\sum_{k=1}^{n} k^3$.

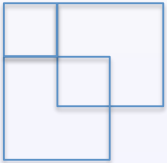
Alternative geometrical proof

Another way of determining the sum of cubes by means of triangular numbers.
The idea is that the cube of k is written as k times the squares k by k.

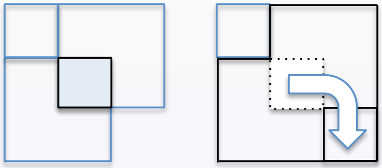
Alternative geometrical proof

Another way of determining the sum of cubes by means of triangular numbers.
The idea is that the cube of k is written as k times the squares k by k.

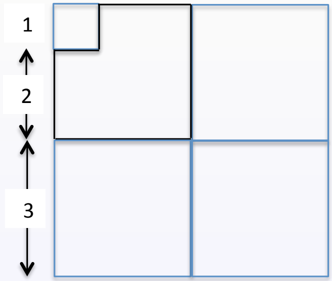
Summation of the first two cubes:



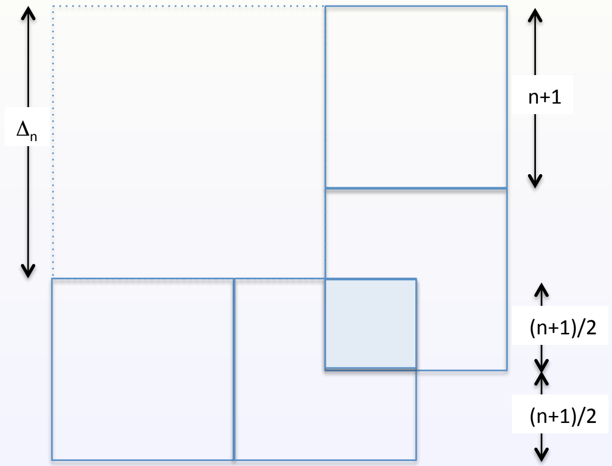
Rearranging 2^3



Adding the term 3^3



General iteration of the summation of cubes



Master Theorem

The cost analysis of the divide-and-conquer paradigm leads to the following recurrence equation. $a \geq 1$ et $b > 1$.

- $f(n) = \Theta(1)$ if $n \leq n_0$
- $f(n) = a.f(\frac{n}{b}) + c(n)$ if $n > n_0$

We give below the general formulation for solving this equation:

1. if $c(n) \in \mathcal{O}(n^{log_b a - \epsilon})$ then $f(n) \in \Theta(n^{log_b a})$
2. if $c(n) \in \Theta(n^{log_b a})$ then $f(n) \in \Theta(n^{log_b a} log(n))$
3. if $c(n) \in \Omega(n^{log_b a + \epsilon})$ and if $a.c(n/b) \leq kf(n)$ for some constant $k < 1$ then, $f(n) \in \Theta(c(n))$

where ϵ is a positive real number.

Analysis of simplified (regular) cases

We assume that n is a power of b (in other words, it can be perfectly divided by b until reaching the value 1).

- $f(1) = 1$
- $f(n) = a.f(\frac{n}{b}) + c$ (c is a positive constant)

- $f(1) = 1$
- $f(n) = a.f(\frac{n}{b}) + n$

Analysis of the simplest case

Let the function f be specified by the simple linear recurrence. Then the value of f on any argument n is given by

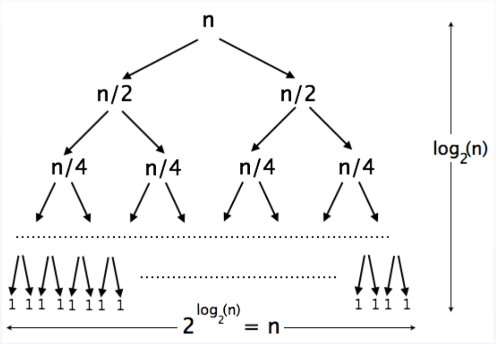
$$\begin{aligned} f(n) &= (1+\log_b n)\cdot c & & \text{if } a = 1 \\ &= \frac{1-a^{\log_b n}}{1-a}\cdot c \approx \frac{c}{1-a} & & \text{if } a < 1 \\ &= \frac{a^{\log_b n}-1}{a-1}\cdot c & & \text{if } a > 1 \end{aligned} \tag{1}$$

proof

We start by expanding the specified computation—replacing occurrences of $f(\bullet)$. Once we discern the pattern, we jump to the general form.

$$\begin{array}{rcccc}
f(n) & = & af(n/b)+c & & \\
 & = & a\left(af(n/b^2)+c\right)+c & = & a^2f(n/b^2)+(a+1)c \\
 & = & a^2\left(af(n/b^3)+c\right)+(a+1)c & = & a^3f(n/b^3)+(a^2+a+1)c \\
 & & \vdots & & \vdots \\
 & = & \left(a^{\log_b n}+\cdots+a^2+a+1\right)c & &
\end{array} \qquad (2)$$

Picturial view for $a = b = 2$



Linear cost instead of constant

We simplify the problem in two ways, in order to avoid calculational complications (such as floors and ceilings) that can mask the principles that govern our analysis.

1. We employ a very simple function g: We focus on the case $g(n) = n$.
   We assume that the argument n to functions f and g is a power of b[1].
2. We consider $c = 1$.

Removing these assumptions would significantly complicate our calculations, but it would not change the reasoning!

[1] This allows us to concentrate on the general unfolding of the recurrence without worrying about floors and ceilings

proof

Let the function f be specified by the general linear recurrence. Then, the value of f on any argument n is given by

$$f(n) \;=\; a^{\log_b n} f(1) \;+\; \left(\sum_{i=0}^{\log_b(n)-1} (a/b)^i\right) n$$

When $a > b$, the behavior of $f(n)$ is dominated by the first term:

$$a^{\log_b n} \cdot f(1) \;=\; n^{\log_b a}$$

When $a < b$, the behavior of $f(n)$ is dominated by the second term:

$$n \cdot \sum_{i=0}^{\log_b(n)-1} (a/b)^i \;=\; \frac{\left(1 \;-\; (a/b)^{\log_b(n)}\right)}{1-(a/b)} \cdot n \;\approx\; \frac{b}{b-a} \cdot n$$

proof

We expose the algebraic pattern created by the recurrence "unfolding". As before, once we discern this pattern, we jump to the general form (which can be verified via induction). $f(n)$

$$\begin{array}{ccccc}
= & af(n/b) + n & & \\
= & a\left(af(n/b^2) + n/b\right) + n & = & a^2 f(n/b^2) + (an/b + \\
= & a^2\left(af(n/b^3) + n/b^2\right) + (a/b+1)n & = & a^3 f(n/b^3) + (a^2/b^2 + a/b \\
 & \vdots & & \vdots \\
= & a^{\log_b n} f(1) + \left(\sum_{i=0}^{\log_b(n)-1} (a/b)^i\right) n & &
\end{array}$$

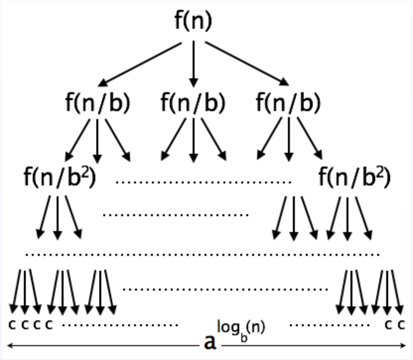
proof

We thus see that solving the more general recurrence requires only augmenting the solution to the simple recurrence by “appending” to the simple solution a geometric summation whose base is the ratio a/b.

Picturial view for any a and b

$f(n) = a.f(\frac{n}{b}) + c(n)$

Solve this problem for a (positive) multiplicative function c defined on the successive powers of b.



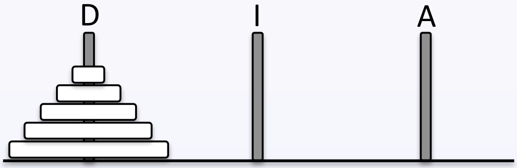
Interpretation of the master Theorem

Intuitively, the theorem tells us that $F(n)$ is determined by the dominance of one of the functions: granularity of the partitioning and merge process.

if $n^{log_b a}$ dominates, then, the solution is in $\Theta(n^{log_b a})$, if this is the contrary, the solution is in $\Theta(f(n))$.
if it is well-balanced, the solution is in $\Theta(f(n).log(n))$

Non-linear recurrences: example Hanoi's Towers

A set of *n* disks of decreasing diameters initially stacked on one of three pegs.
Problem definition. The goal is to transfer the entire tower from a given peg to another fixed one, moving only one disk at a time and never moving a larger disk on top of a smaller one.



The main question is to determine the *best* way to realize this operation (which means in a minimum number of moves).

Lower Bound

Let call H_n the number of moves required to solve the puzzle with n disks. **What is the minimum number of moves?**
When there is only one disk, there is only one move: $H_1 = 1$, with two disks, at least 3 moves are necessary: $H_2 = 3$.

Let us prove $H_n \geq 2H_{n-1} + 1$. Indeed, looking at the largest disk, the $n-1$ others must be on a single peg, which required H_{n-1} to put them here. Then, the largest disk should be moved, and again move the $n-1$ others on the target peg.
Thus, we have the following recurrence to solve: $H_n = 2H_{n-1} + 1$ $(n \geq 2)$ with $H_1 = 1$.

There exists a closed formula: The first ranks give us an insight of the solution (1, 3, 7, 15, 31, ...).
We guess $H_n = 2^n - 1$ for $n \geq 1$.

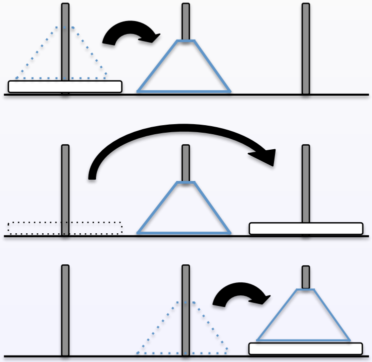
Compute the cost

- **Basis** is straightforward since $H_1 = 2^1 - 1 = 1$.
- **Induction step**
  $H_n = 2H_{n-1} + 1$ where $H_{n-1} = 2^{n-1} - 1$,
  thus, $H_n = 2(2^{n-1} - 1) + 1 = 2^n - 1$ and we are done.

Notice that this expression can also be obtained directly as the sum of a geometric series:
$H_n = 2H_{n-1} + 1$
$= 2(2H_{n-2} + 1) + 1$
$= \sum_{j=0}^{n-1} 2^j$
$= \frac{1-2^n}{1-2} = 2^n - 1$

The classical (recursive) solution

The natural method is recursive. It consists in moving the $n-1$ top disks on the intermediate peg, then, put the largest one on the target peg, and moving again the $n-1$ disks on top of it.



formal algorithm

input: an integer n, three pegs indexed as D (departure), A (arrival) and I (intermediate). All the disks are stacked on peg D.
output: The disks are stacked on peg A.

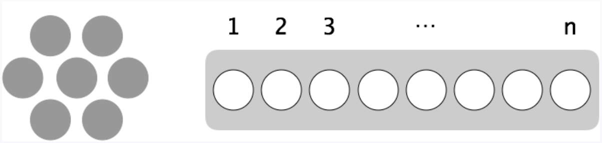
If $(n \neq 0)$ then

- Hanoi(n-1,D,I,A)
- move disk from D to A
- Hanoi(n-1,I,A,D)

Analysis. It is easy to compute the number of moves, using the same recurrence equation as for the lower bound:
$H_n = 2^n - 1$.

The Token Game

Consider a bank with n circle positions numbered from 1 to n and n tokens. Initially, the bank is empty.



The game consists in determining the process to fill the bank with the n tokens, putting or removing one token at a time according to one of the two following constraints.

- **Rule 1.** Position 1: Put a token if it is empty or remove it.
- **Rule 2.** Position next to the first empty position (i.e. on the right): Put a token if the position is empty or remove it.

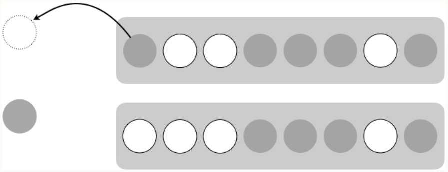


Figure: Rule 1: Position 1 contains a token, thus, remove it.

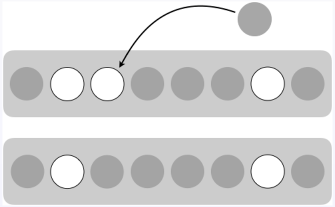


Figure: Rule 2: The position next to the first idle position (i.e. position 3 here) is idle, thus, put a remaining token.