SOLVING RECURRENCES

Denis TRYSTRAM

Lecture notes **Maths for Computer Science** – *MOSIG 1 – 2014*

1 Principle

Mathematical induction is used for proving that a statement involving integers is true. The method is called *recurrence*.

- **Basis.** Solve the statement for the smallest values of n.
- **Induction step.** Prove the statement for n assuming it is correct for all $k < n$.

2 Towers of Hanoi

This introductory problem comes from the famous french mathematician Edouard Lucas (1883). We are given a set of n disks of decreasing diameters initially stacked on one of three pegs.

The goal is to transfer the entire tower from this peg to another one, moving only one disk at a time and never moving a larger disk on top of a smaller one.

The questions of this puzzle is to determine the best way to realize this operation (meaning with a minimum number of moves).

The number of moves can easily be determined:

- When there is only one disk, there is only one move: $H_1 = 1$, with two disks, 3 moves are necessary: $H_2 = 3$.
- On the first hand, $H_n \leq 2H_{n-1} + 1$ because there is a natural recursive method which consists in moving the $n-1$ top disks, then, the largest one, and putting again the $n-1$ disks on top of it. This is a upper bound since may be there exists a better method...
- on the second hand, $H_n \geq 2H_{n-1} + 1$. 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 the $n-1$ others on a single peg.

All together, we have the following recurrence to solve: $H_n = 2H_{n-1} + 1$ 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 $n \geq 1$.

The basis is straightforward, the induction step follows:

$H_n = 2H_{n-1} + 1$ 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 be obtained directly by applying successively the induction steps:

$H_n = 2H_{n-1} + 1 = 2(2H_{n-2} + 1) + 1 = 2(2(2H_{n-3} + 1) + 1) + 1 = ...$
$= \sum_{j=0}^{n-1} 2^j = \frac{1-2^n}{1-2} = 2^n - 1$

2.1 Extension

Let us study the problem for a larger number of pegs (k). Let Hanoi(n,k) denotes this problem. The problem studied in the previous section corresponds to Hanoi(n,3).

Now we are interested in the problem where the number of pegs is note fixed (let call is k), in particular, what is the minimum number of moves that can be achieved if this number is as large as needed.

It is clear that for $k = n + 1$, the number of move is linear in n (just move each disk on a different peg (which requires $n - 1$ moves), move the largest one (1 move) and put them one after the other on top of the largest ones ($n - 1$ moves), thus, a total of $2n - 1$ moves.

Actually, it is the absolute lower bound.

An interesting question is if this bound can be achieved for less pegs than $n + 1$. The answer is "yes". This can be obtained by dividing the n disks into blocks of size $\sqrt{n}$ and consider $k = 2\sqrt{n}$ pegs.

3 Some recurrence on Fibonacci numbers

Fibonacci numbers are defined by the following numerical progression: $n \geq 2$ $F(n) = F(n-1) + F(n-2)$, $F(0) = F(1) = 1$. These numbers have nice properties, like the following ones.

- The relation $F(n-1).F(n) = \sum_{k=0}^{n-1} F^2(k)$ (for $n \geq 1$) can be proved by using a geometric argument (see figure 1). Show it by induction.
- $F(n+1)^2 = 4.F(n).F(n-1) + F(n-2)^2$ for $n \geq 2$
- Show the following Cassini's identity: $F(n-1).F(n+1) = F^2(n) + (-1)^{n+1}$ for $n \geq 1$.

  Here is the proof:

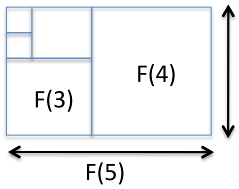
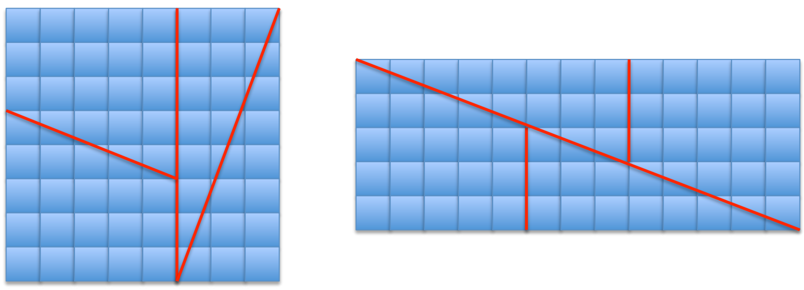


Figure 1: Geometric interpretation of the relation $F(4).F(5) = F^2(0) + F^2(1) + F^2(2) + F^2(3) + F^2(4)$.



$F(n).F(n+2) = F(n)(F(n+1)+F(n))$ by definition of the Fibonacci progression

$= F(n).F(n+1) + F(n)^2$

from the recurrence hypothesis, we have

$F^2(n) = F(n-1).F(n+1) - (-1)^{n+1} = F(n-1).F(n+1) + (-1)^{n+2}$
Thus,

$F(n).F(n+2) = F(n).F(n+1) + F(n-1).F(n+1) + (-1)^{n+2}$

$= F(n+1)(F(n)+F(n-1)) + (-1)^{n+2}$ and again, since $F(n)+F(n-1) = F(n+1)$

$= F^2(n+1) + (-1)^{n+2}$

- The previous result (Cassini's identity) is the basis of a geometrical paradox (one of the favorite puzzle of Lewis Carroll). Consider a chess board and cut it into 4 pieces as shown in figure 3, then reassemble them into a rectangle.

  The surface of the square is F_n^2 while the rectangle is $F_{n+1}.F_{n-1}$. The Cassini identity is applied for $n = 8$. The paradox comes from the wrong representation of the diagonal of the rectangle which does not coincide with the hypothenus of the rectangle triangles of sides F_{n-1} and F_{n-2}. In other words, it always remains (for any n) an empty space (corresponding to the unit size of the basic case of the chess

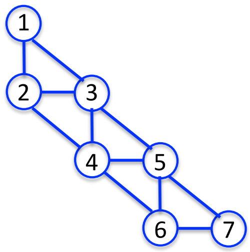


Figure 2: Counting paths from node 1 to node n ($n = 7$)

board). The greater n, the better the paradox because the surface of this basic case becomes more tiny.

- $F(n)$ is the number of paths from node 1 to n in the following family of graphs of figure 2. Show how this number is related to Fibonacci's numbers.

4 Recurrences on graphs

Let start by proving a well-known statement on trees. Let first recall that Trees are defined as connected graphs without cycles. The statement to prove it that any tree has exactly $n - 1$ edges.

This is done by induction on n as follows:

- **Basis.** A tree with only one vertex has no edge.
- **Induction step.** Assuming that any tree of order n has $n - 1$ edges, we consider a tree of order $n + 1$, it is easy to see (by contradiction) that there exists at least one leaf (a vertex with no successor). Then, if we remove the edge adjacent to this leaf, we obtain a tree of order n (the graph is still connected and no cycles have obviously been added). This tree has $n - 1$ edges, thus, together with the removed edge, the original tree has n edges.

Similarly to the last example, most graph problems that use recurrences are done on the order of the graph n (number of vertices). However, there are some recurrences based on induction on m (the number of edges) like the basic proof of existence of eulerian cycles in graphs whose vertices have all an even degree.

5 More complex recurrences

Now, we are going to solve a recurrence equation which comes from the divide and conquer strategies in algorithm design. Counting the number of operations leads to expressions of the following type:

$T(n) = aT(\frac{n}{b}) + f(n)$ for $n \geq 2$ where a and b are two fixed integers

$T(1) = 1$

This may be interpreted as subdividing a problem of size n into a subproblems of sizes $\frac{n}{b}$. Function f corresponds to the partition operations (if any) and the recomposition of the partial solutions.

The general form is hard to compute, let assume that n is a power of b: $n = b^k$. Thus, the equation becomes $T(b^k) = aT(b^{k-1}) + f(b^k)$.

If we set $t_k = T(b^k)$ (and $t_0 = 1$), we obtain a simpler (linear) form as follows:

$t_k = a^k + \sum_{j=0}^{k-1} a^j f(b^{k-j})$

where $k = log_b n$, as $a^{log_b n} = n^{log_b a}$, the equation can be written as:

$T(n) = n^{log_b a} + \sum_{j=0}^{log_b n-1} a^j f(\frac{n}{b^j})$

Applied to the fusion sort (where $a = 2$ and $b = 2$ and $f(n) = n - 1$), the equation leads to:

$T(n) = n + \sum_{j=0}^{log_2 n} 2^j(\frac{n}{2^j} - 1) = nlog_2 n + 1$

Solving the general form requires more sophisticated techniques. A closed form can be obtained for the particular case of f (multiplicative functions, which verify $f(x_1.x_2) = f(x_1).f(x_2)$).

6 Josephus' problem

The problem comes from an old story reported by Flavius Josephus during the Jewish-Roman war between in the first century. Flavius was among a band of 41 rebels trapped in a cave by the roman army. Preferring suicide to capture, the rebels decided to form a circle and proceeding around to kill every third remaining person until no one was left. As Josephus had some skills in Maths and wanted none of this suicide non-sense, he quickly calculated should stand at the end of the process.

Definition. Given n successive numbers in a circle. The problem is to determine the survival number (denoted by $J(n)$) in the process of removing every second remaining number starting from 1 (see figure 3).

In particular, we are going to determine if there exists a closed formula. Guessing the answer sounds not obvious. We need to better understand the progression.

Property 1. $J(n)$ is odd

Proof. This is straightforward since the first tour removes all even numbers! See figure 4.

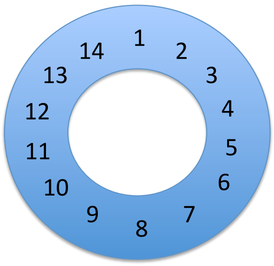


Figure 3: Initial situation for the Josephus process.

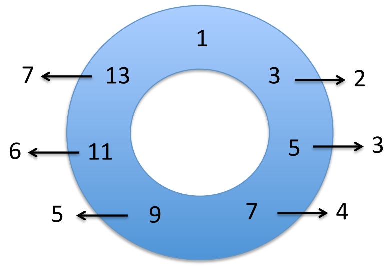


Figure 4: First step of the process (n is even).

Property 2. (even numbers) $J(2n) = 2J(n) - 1$
Proof. This is a simple generalization of the previous property.

From this, we deduce $J(2^m) = 1$ for all m.
Let us turn to odd numbers.

Property 3. (odd numbers) $J(2n + 1) = 2J(n) + 1$

We can compute easily the first ranks. It turns out that the progression is composed of grouped terms starting at each power of 2. Let $n = 2^m + k$, the rule within each group m is to start at 1 and increase by 2 the successive numbers $(0 \leq k < 2^m)$. Let prove it by recurrence on m.

Property 4. $J(2^m + k) = 2k + 1$
Proof.

- **Basis.** $m = 0$, thus $k = 0$ and $J(1) = 1$
- **Induction step.** Suppose the formula holds for any integer lower than $n = 2^m + k$. Since there are two expressions for $J(.)$, we distinguish the cases k whether is even and k is odd:

- If k is even, then, $2^m + k$ is even, and we can write:
  $J(2^m + k) = 2J(2^{m-1} + \frac{k}{2}) - 1$
  by induction hypothesis, $J(2^{m-1} + \frac{k}{2}) = 2\frac{k}{2} + 1 = k + 1$
  Thus, $J(2^m + k) = 2(k + 1) - 1 = 2k + 1$.
- If k is odd, the proof is similar:
  $J(2^m + k) = 2J(2^{m-1} + \lfloor \frac{k}{2} \rfloor) + 1 = 2\lfloor \frac{k}{2} \rfloor + 1 = 2k + 1$.

We can even go one step further with this problem by remarking that powers of 2 play an important role. Let us use the radix 2 representation of n and $J(n)$:
$n = \sum_{j=0}^{j=m} b_j.2^j = b_m.2^m + b_{m-1}.2^{m-1} + ... + b_1.2 + b_0$
$n = (1b_{m-1}...b_1b_0)_2$ since by definition of m $b_m = 1$
$k = (0b_{m-1}...b_1b_0)_2$ since $k < 2^m$
Thus, using the closed formula for $J(n)$:
$J(n) = (b_{m-1}...b_0b_m)_2$.

In other words, the solution is obtained by a simple shift of the binary representation of n. Applied to $n = 41 = (101001)_2$ Josephus Flavius was able to determine the last position in few seconds: $(010011)_2 = 19$.