Introductory examples

Many of the calculations in this course are values that are worked out *successively*: each number is calculated from the ones already known.

- Edges in K_n: the kth vertex added adds $k-1$ new edges.
- Vertices in Q_n: Q_n has twice the vertices of Q_{n-1}
- $n!$ is worked out from $(n-1)!$ via $n! = n\,(n-1)!$
- Permutations of k objects out of n: $P(n,k) = (n-k+1)\,P(n,k-1)$
- Derangements: $D_n = n\,D_{n-1} + (-1)^n$.

For each of these there is a recurrence relation that relates one value in the sequence to the previous values.

Formal definitions

Sequence is an infinite list of numbers, written as subscripted variables with consecutive subscripts. For example $a_1, a_2, a_3, \ldots$ and $x_0, x_1, x_2, \ldots$. Common abbreviation: $\{a_k\}_{k=1}^{\infty}$ and $\{x_i\}_{i=0}^{\infty}$.

Recurrence relation for a sequence is an equation relating each value in the sequence to others in a well-defined way. For example

$$a_n = 3a_{n-1}\ , \quad x_k = x_{k-1} + k\ , \quad D_n = n\,D_{n-1} + (-1)^n$$

Initial conditions for a recurrence relation are values given for enough of the first values in the sequence, so that all subsequent values can be determined by the recurrence relation.

Example

Consider the recurrence relation $a_n = 3a_{n-1}$ for the sequence $a_0, a_1, a_2, \ldots$. By itself the recurrence relation just gives relationships between the subscripted values:

$$a_2 = 3a_1 = 9a_0 \quad ; \quad a_{n+k} = 3^k \, a_n$$

Now specify initial conditions $a_0 = 5$. Then we use the recurrence relation to find subsequent values of a_n by putting in $n = 1, 2, 3, \ldots$:

$$\begin{array}{cc} n = 1 & a_1 = 3a_0 = 15 \\ n = 2 & a_2 = 3a_1 = 45 \\ n = 3 & a_3 = 3a_2 = 135 \\ \vdots & \vdots \end{array}$$

Example (Fibonacci)

The Fibonacci sequence starts 1, 1 and each subsequent term is the sum of the two preceeding terms: $1, 1, 2, 3, 5, 8, 13, 21, \ldots$. The recurrence relation is $f_n = f_{n-1} + f_{n-2}$ and initial conditions are $f_1 = 1$, $f_2 = 1$
Fibonacci wrote about the sequence in relation to a population dynamics problem. We will look at this later.

Example

An investment of \$10000 is set up so that it attracts 0.5% interest at the end of each month. Write a recurrence relation with initial conditions for a_n, the balance after n months, and find the balance after 12 months.

Example

As in the above example, except I get charged a fee of \$10 per month after the interest is taken out.

Recurrence relation:	$a_n = 1.005\, a_{n-1} - 10$
Initial conditions:	$a_0 = 10000$

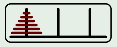
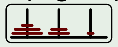
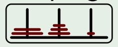
Analysis of algorithms

The analysis of the efficiency of algorithms often relies on Recurrence relations.
The next two examples apply recurrence relations to algorithms:

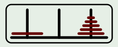
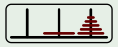
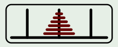
- the solution to the tower of Hanoi puzzle
- the merge sort algorithm

Example (Tower of Hanoi)

Description: The Tower of Hanoi is a puzzle

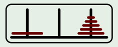
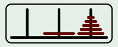
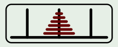
- three pegs with n disks of decreasing size on one peg 
- a legal move takes one disk from peg to peg keeping each stack in decreasing order (small on big)  $\rightarrow$ 
- Goal: move the stack to another peg in minimum moves

To move the stack $\{1, 2, \ldots, n-1, n\}$ from A to B:

- We must move disk n from A to B — can't!
- First move stack $\{1, 2, \ldots, n-1\}$ aside (to C) 
- Then move disk n from A to B 
- Finally move stack $\{1, 2, \ldots, n-1\}$ from C to B. 

Example (Tower of Hanoi)

Description:
Algorithm: Label the disks 1 (tiny) to n (huge) and the pegs A, B and C.
To move the stack $\{1, 2, \ldots, n-1, n\}$ from A to B:

- We must move disk n from A to B — can't!
- First move stack $\{1, 2, \ldots, n-1\}$ aside (to C) 
- Then move disk n from A to B 
- Finally move stack $\{1, 2, \ldots, n-1\}$ from C to B. 

Analysis: Let M_n be the number of moves required to move a stack $\{1, 2, \ldots, n\}$ from peg to peg.
The recurrence relation with initial conditions is

$$M_n = M_{n-1} + 1 + M_{n-1} = 2M_{n-1} + 1\ ; \quad M_1 = 1$$

Example (Merge sort)

Suppose we apply the Merge sort algorithm to a list of n elements where $n = 2^k$. This involves

- Sort the first $n/2 = 2^{k-1}$ elements to get a list A
- Sort the last $n/2 = 2^{k-1}$ elements to get a list B
- Merge lists A and B

Merging two lists of length m requires, in the worst case, $2m - 1$ comparisons. Consequently, if C_k is number of comparisons required (in the worst-case) to merge-sort a list of 2^k elements, then

$$C_k = 2C_{k-1} + 2^k - 1$$

Since $C_1 = 1$, we can find values for $C_2, C_3, C_4, \ldots$

Example (Balanced Trees)

When we were balancing trees (according to heights of subtrees) the intention was that this gave us a relatively efficient distribution of vertices in the tree. In particular, close to the minimal height of the complete binary tree.
To appraise this, we try to construct the "worst case" balanced tree. This could be done by either

1. Given n vertices, construct a balanced tree of maximum height
2. Given height k, construct a balanced tree of height k with minimal vertices

We can solve the second by defining v_k be the minimum number of vertices in a balanced tree of height k.

Notation

If we have a recurrence relation (RR) for a sequence $\{a_n\}_{n=0}^{\infty}$,

A solution to the RR is a formula that gives a_n purely in terms of n.

The general solution to the RR is a formula for the sequence that covers all possible choices of initial conditions.

If, in addition to the RR, we have initial conditions given, then

The solution to this problem (RR&IC) is a formula for just one sequence from those described in the general solution, such that it satisfies the initial conditions.

Example

For instance, with the recurrence relation $a_n = 3a_{n-1}$,

- there are many solutions, such as $a_n = 3^n$, $a_n = 0$, $a_n = 7 \times 3^n$.
- The general solution is $a_n = A \times 3^n$.
- With initial conditions $a_0 = 5$, the solution is $a_n = 5 \times 3^n$.

Sums and Products

Not all recurrence relations are easily solved. However, there are certain recurrence relations that we can solve.

The simplest case is where the recurrence relation gives rise to a process that builds up a sum or a product.

Just as with induction, a good strategy is to work out the first few cases, looking for evidence of the process that takes you from one number to the next.

Example (Summation 1)

Recurrence relation $a_n = a_{n-1} + 7$.

Example (Summation 2)

Recurrence relation $s_n = s_{n-1} + 2n$ with initial condition $s_0 = 1$.

Example (Product 1)

Recurrence relation $q_n = \frac{1}{10} q_{n-1}$

Example (Product 2)

Recurrence relation $F_n = nF_{n-1}$ with initial condition $F_0 = 1$.

Example (Product 3)

Recurrence relation $b_n = \frac{n}{2} b_{n-1}$ with initial condition $b_0 = 4$.

Example (Hanoi)

Recurrence relation $M_n = 2M_{n-1} + 1$ with initial condition $M_1 = 1$.

Example (Investment fees)

Recurrence relation $a_n = 1.005a_{n-1} - 10$ with initial condition $M_0 = 10000$.

Example (Merge sort)

Recurrence relation $C_k = 2C_{k-1} + 2^k - 1$ with initial condition $C_1 = 1$.

A very special case

A constant-coefficient homogeneous linear recurrence relation of the form

$$\begin{array}{rrcll} & s_n & = & as_{n-1} & \text{(first order)} \\ \text{OR} & s_n & = & as_{n-1} + bs_{n-2} & \text{(second order)} \\ \text{OR} & s_n & = & as_{n-1} + bs_{n-2} + cs_{n-3} & \text{(third order)} \\ \text{etc.} & & & & \end{array}$$

where $a, b, c, \ldots$ are constants.

We will call these linear recurrence relations.

Solution of first-order case

We have seen the first-order case $s_n = as_{n-1}$ in examples before.
It gives rise to a product

$$s_n = s_0 \left(\prod_{i=1}^{n} a \right) = s_0 a^n$$

Which means $\{s_n\}_{n=0}^{\infty}$ is a geometric sequence.

It is useful to view

$$1, a, a^2, a^3, a^4, \ldots$$

as a "basic solution" and any other solution is a multiple of this.

Solution of second-order case

A second-order linear RR $s_n = as_{n-1} + bs_{n-2}$ does not lead to a product as occurred in the first-order case.

An example is the Fibonacci Recurrence Relation $f_n = f_{n-1} + f_{n-2}$. Which has solution

$$1, 1, 2, 3, 5, 8, 13, 21, \ldots$$

Each term is **NOT** a multiple of previous terms.

However, the first step towards solving second-order equations is to note that *certain* solutions **are** geometric progressions — these will then feature in the general solution.

Basic solutions for the second-order case

Theorem

Suppose $s_n = as_{n-1} + bs_{n-2}$ *is a second-order linear recurrence relation. Then the geometric series* $1, r, r^2, r^3, \ldots$ *is a solution if, and only if,* r *is a solution of the quadratic equation* $x^2 = ax + b$

Proof.

If $s_n = r^n$ is a solution, then with $n = 2$ we have $r^2 = ar + b$.
Conversely, if $r^2 = ar + b$ then the sequence $s_n = r^n$ has

$$s_n = r^n = r^2 r^{n-2} = (ar + b)r^{n-2} = ar^{n-1} + br^{n-2} = as_{n-1} + bs_{n-2} \qquad \square$$

The equation $x^2 = ax + b$ is known as the characteristic equation.

The Fibonacci recurrence relation (FRR)

$$f_n = f_{n-1} + f_{n-2}$$

is linear, and we can solve its characteristic equation $x^2 = x + 1$ using the quadratic formula:

$$x^2 - x - 1 = 0 \quad \Rightarrow \quad x \;=\; \frac{1 \pm \sqrt{5}}{2} \;\approx\; 1.618, -0.618$$

By the theorem, there are two exponential solutions to the FRR:

$$\begin{array}{rllllll}
\text{Solution 1:} & 1, & \frac{1+\sqrt{5}}{2}, & \left(\frac{1+\sqrt{5}}{2}\right)^2, & \left(\frac{1+\sqrt{5}}{2}\right)^3, & \left(\frac{1+\sqrt{5}}{2}\right)^4, & \dots \\
\approx & 1, & 1.618, & 2.618, & 4.236, & 6.854, & \dots \\
\text{Solution 2:} & 1, & \frac{1-\sqrt{5}}{2}, & \left(\frac{1-\sqrt{5}}{2}\right)^2, & \left(\frac{1-\sqrt{5}}{2}\right)^3, & \left(\frac{1-\sqrt{5}}{2}\right)^4, & \dots \\
\approx & 1, & -0.618, & 0.382, & -0.236, & 0.146, & \dots
\end{array}$$

Both of these sequences satisfy the FRR, but neither is the Fibonacci sequence. They are not integers!

However, we can get some integer solutions by manipulating the two solutions above.
Adding the two solutions above gives

$$2, 1, 3, 4, 7, \dots$$

which is an integer solution to the FRR. However this is not the Fibonacci sequence either.
Subtracting gives approximately

$$0, 2.236, 2.236, 4.472, 6.708, \dots \quad \text{or exactly:} \quad 0, \sqrt{5}, \sqrt{5}, 2\sqrt{5}, 3\sqrt{5}, \dots$$

which is $\sqrt{5}$ times the Fibonacci sequence. Consequently

$$f_n = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right]$$

This hints at a strategy for solving second-order linear RRs:

- Find two exponential sequences that are solutions to the RR
- Find some combination of the two that matches the initial values

Theorem

Suppose $s_n = as_{n-1} + bs_{n-2}$ is a second-order linear recurrence relation for a sequence $s_0, s_1, s_2, \ldots$. If the quadratic equation $x^2 = ax + b$ has two distinct (real) solutions r_1, r_2 then for any constants A, B, the sequence

$$s_n = Ar_1^n + Br_2^n$$

is a solution to the recurrence relation, and every solution is of this form.

Sketch proof.

By the theorem, both $s_n = r_1^n$ and $s_n = r_2^n$ are solutions. Also, the set of solutions is closed under sums and constant multiples, so $s_n = Ar_1^n + Br_2^n$ is a solution for any A, B.

If $u_0, u_1, u_2, \ldots$ was a sequence that was a solution, then solve $A + B = u_0$ and $Ar_1 + Br_2 = u_1$ simultaneously to find A, B. Then $t_n = u_n - Ar_1^n - Br_2^n$ will be a solution with $t_0 = t_1 = 0$. Then $t_n = 0$ for all n, so $u_n = Ar_1^n + Br_2^n$. □