Analyze of recursive algorithms

- The most important upside of a recursive expression is the fact that it is natural and compact, without hiding the essence of algorithm through details of implementation.
- On the other hand, recursive calls must be used with care, because they also require computer resources (time and memory).
- Analysis of an recursive algorithm implies solving a system of recurrences.

Recursive relations

- When an algorithm contains a recursive call to itself, its time of execution can be described with a reaccurence.
- A *reacurrence* is an equation or inequation that describes entire time of execution of a problem of n size with the help of times of execution for input data of small size.
- There exist mathematical tools for solving reacurrence problems and for obtaining margins of algorithm performances.

Equation characteristic method

There a few types of reacurrences:

- Linear homogeneous reacurrences
- Linear nonhomogeneous reacurrences
- Nonlinear reacurrences

Linear homogeneous reacurrences

We will consider linear homogeneous reacurrence of form:

$$a_0 t_n + a_1 t_{n-1} + \ldots + a_k t_{n-k} = 0 \quad (1)$$

where t_i are the values we are looking for and coefficients a_i are constants.

We will search for solutions of form:

$$t_n = x^n$$

where x is a constant (unknown, for now)

Linear homogeneous reacurrences

We try this solution (1) and obtain:

$$a_0 x^n + a_1 x^{n-1} + \ldots + a_k x^{n-k} = 0$$

Solutions of this equation are either trivial(x = 0), which we are not interested in, or solutions for the equation:

$$a_0 x^k + a_1 x^{k-1} + \ldots + a_k = 0 \ (2)$$

which is ***characteristic equation*** of reacurrence (1).

Linear homogeneous reacurrences

Assuming that those k roots $r_1, r_2, \ldots, r_k$ of this characteristic equation are distinct, any linear combination

$$t_n = \sum_{i=1}^{k} c_i r_i^n$$

is a ***solution*** of reacurrence (1), where constants $c_1, c_2, \ldots, c_k$ are determined by initial conditions.

It must be mentioned that (1) has solutions only of this form.

Example

Reacurrence that defines Fibonacci sequence:

$$t_n = t_{n-1} + t_{n-2} \qquad n \geq 2$$

and $t_0 = 0, t_1 = 1$

We can rewrite this reacurrence in form:

$$t_n - t_{n-1} - t_{n-2} = 0$$

which is ***characteristic equation***

$$x^2 - x - 1 = 0$$

with roots $r_{1,2} =$

Example

Reacurrence that defines Fibonacci sequence:

$$t_n = t_{n-1} + t_{n-2} \qquad n \geq 2$$

and $t_0 = 0, t_1=1$

We can rewrite this reacurrence in form:

$$t_n - t_{n-1} - t_{n-2} = 0$$

which is chracterstic equation

$$x^2 - x - 1 = 0$$

with roots $r_{1,2}$ = (1+sqrt5)/2, (1- sqrt5)/2

Example

General solution is of form:

$$t_n = c_1 r_1^n + c_2 r_2^n$$

Inputting initial conditions, we obtain

$c_1 + c_2 = 0,$ $\qquad n = 0$

$r_1 c_1 + r_2 c_2 = 1,$ $\qquad n = 1$

where we can determine

Example

General solution is of form

$$t_n = c_1 r_1^n + c_2 r_2^n$$

Inputting initial conditions, we obtain

$c_1 + c_2 = 0,$ $\quad n = 0$

$r_1 c_1 + r_2 c_2 = 1,$ $\quad n = 1$

where we can determine c_1=1/sqrt5, c_2 =-1/sqrt5

Example

Solve the reacurrence

$$\boldsymbol{t_n - 3t_{n-1} - 4t_{n-2} = 0,}$$

where $n \geq 2$, and $t_0 = 0$, $t_1 = 1$

r1= 4, r2=-1

Linear homogeneous reacurrences with multiple roots

- What do we do when characteristic equation's solution are not distinct?
- We can show that, if *r* a root of multiplicity *m* of characteristic equation, then

$$t_n = r^n,\ t_n = nr^n,\ t_n = n^2r^n,\ \ldots,\ t_n = n^{m-1}r^n$$

are solutions for reaccurence (1).

- General solution for this kind of reaccurence a linear combination of these terms and of terms that came from other roots of the characteristic equation.
- Again, must be determined exactly *k* constants from initial conditions.

Example

Solve the reaccurence

$$t_n = 5t_{n\text{-}1} - 8t_{n\text{-}2} + 4t_{n\text{-}3} ,$$

where $n \geq 3$, and $t_0 = 0$, $t_1 = 1$, $t_2 = 2$

Example

Solve the reaccurence

$$t_n = 5t_{n-1} - 8t_{n-2} + 4t_{n-3} \text{ ,}$$

where $n \geq 3$, and $t_0 = 0$, $t_1 = 1$, $t_2 = 2$

$$t_n = c_1 1^n + c_2 2^n + c_3 n 2^n$$

$c_1 = -2$, $c_2 = 2$, $c_3 = -1/2$

$t_n = -2\,(1^n) + 2\,(2^n) + (-1/2)(n2^n)$

Linear nonhomogeneous reacurrences

We consider now reacurrences of more general form

$$a_0 t_n + a_1 t_{n-1} + \ldots + a_k t_{n-k} = b^n p(n) \qquad (3)$$

where b is a constant, and $p(n)$ is a polynom in n of degree d.

Linear nonhomogeneous reacurrences

We can show that, for solving (3), is enough to take the following characteristic equation:

$$(a_0x^k + a_1x^{k-1} + \ldots + a_k)(x-b)^{d+1} = 0 \quad (4)$$

Once this equation is obtained, we proceed as if in case of homogeneous reacurrences.

Linear nonhomogeneous reacurrences

For example, there can be such reacurrence:

$$t_n - 2t_{n-1} = 3^n$$

In this case, $b = 3$ and $p(n) = 1$, a polynom of degree 0.

Characteristic equation is:

$(x-2)(x-3) = 0$ with roots $r_1 = 2$, $r_2 = 3$

General solution will be:

$$t_n = c_1 2^n + c_2 3^n$$

Linear nonhomogeneous reacurrences

Solve reacurrences:

1. $t_n - 2t_{n-1} = 2^n$
2. $t_n - 2t_{n-1} = n3^n$
3. $t_n - t_{n-1} = n$

Change of variable

We will analyze reacurrences of form:

$$\boldsymbol{T(n)= aT(n/b)+ f(n)} \qquad \textbf{(5)}$$

where $a \geq 1$ and $b>1$ are constants, and $f(n)$ is an asymptotically positive function.

Change of variable

Reacurrence (5) describes execution time of an algorithm that splits a problem of size n in a subproblems, each of size n/b, where a and b are positive constants.

Those a subproblems are solved recursively, each in time of $T(n/b)$.

The cost of spliting a problem and combining the results of the subproblems is described by the fucntion $f(n)$ (Meaning, using the notation $f(n)=D(n) + C(n)$).

From techincal view, reacurrence is not, actually, well defined, because n/b may not be whole number.

Change of variable

Sometimes, using change of variable, we can solve reacurences of type (5).

Further, notation $\boldsymbol{T(n)}$ will be the general term of reacurrence and with $\boldsymbol{t_k}$ the term of the new reacurrence obtained using a change of variable.

Assume that, for the start, $\boldsymbol{n}$ is a power of $\boldsymbol{b}$.

Example

Let reacurrence $T(n) = 4T(n/2) + n$, $n > 1$ where we replace n with 2^k, note $t_k = T(2^k) = T(n)$ and obtain

$$t_k = 4t_{k-1} + 2^k$$

$$t_k - 4t_{k-1} = 2^k$$

Characteristic equation of this linear reacurrence is:

$$(x-4)(x-2) = 0 \text{ with } r_1 = 4 \text{ şi } r_2 = 2 \text{ so,}$$

$$t_k = c_1 4^k + c_2 2^k.$$

We replace back k with $\log_2 n$ and obtain

$$T(n) = c_1 4^{\log n} + c_2 2^{\log n}$$

$$T(n) = c_1 n^2 + c_2 n$$

Example

Solve the reacurrences

1. $T(n) = 2T(n/2) + n, \; n > 1$
2. $T(n) = 8T(n/2) + n, \; n > 1$
3. $T(n) = 9T(n/3) + n^2, \; n > 1$
4. $T(n) = 2T(n/2) + n\log n$

Individual work

- Asymptotic efficiency of algorithms
- ***Asymptotic time of an algorithm execution***
- Asymptotic notations Θ, O , o , Ω, ω