CSI2101-W08- Recurrence Relations

Motivation

- where do they come from
  - modeling
  - program analysis

Solving Recurrence Relations

- by iteration – arithmetic/geometric sequences
- linear homogenous recurrence relations with constant coefficients
- linear non-homogenous …

Divide-&-Conquer Algorithms and the Master Theorem

- solving recurrence relations arising in analysis of divide&conquer algorithms

Recurrence Relations - Motivation

Compound interest

- x% interest each year
- how much do you have in your account after 30 years?
- $a_y = (1+x/100)a_{y-1}$

Rabbit breeding

- one adult pair produces new pair each month
- a pair becomes adult in the second month of its life
- no rabbits die
- $r_m = r_{m-1}+r_{m-2}$
- the Fibonacci sequence

Recurrence Relations - Motivation

The towers of Hanoi

- move a pyramid of discs from one peg to another, using a third peg
- bigger disc cannot be placed on a smaller one
- the algorithm:
  - move the top **n-1** discs from **A** to **C** using **B** (recursively)
  - move the bottom disc from **A** to **B**
  - move the top **n-1** discs from **C** to **B** using **A** (recursively)
- cannot be done any faster:
  - the bottom disc can be moved only after all the discs above it have been moved
- let H_n denote the minimal time to solve the problem with **n** discs
  - then $H_n = H_{n-1}+1+H_{n-1} = 2H_{n-1}+1$

Recurrence Relations - Motivation

The number of binary strings without two consecutive 0s

- how many such strings of form **X1** (the ones that end in **1**)?
  - as many as there are such strings **X** of length **n-1**
- how many of form **X0**?
  - **X** must end in **1** (i.e. **X = Y1**)
  - **Y10** – as many as there are such strings **Y** of length **n-2**
- $c_n = c_{n-1} + c_{n-2}$

The number of binary strings without three consecutive 0s

- X1, Y10, Z100
- $d_n = d_{n-1} + d_{n-2} + d_{n-3}$

Recurrence Relations - Motivation

The number of different ways to parenthesize $x_0*x_1*x_2...*x_{n-1}$

- corresponding to different orders of computing the product
- **$n-1$** ways to choose which will be the last multiplication
  - $(x_0*x_1*...x_{i-1})*(x_i*...x_{n-1})$ for $i=1 ... n-1$
- recursively, if we choose to split at i, the number of different ways is is C_i*C_{n-1-i}
- summing up for all i we get

  $$C_n = \Sigma_{i=1}^{n-1} C_i*C_{n-1-i}$$

- the sequence C_n is called **Catalan numbers**

Solving Recurrence Relations

Difficult in general, we will focus on the easier cases:

Linear homogenous recurrence relation of degree k with constant coefficients:

- $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \ldots + c_k a_{n-k}$
- linear = only a_i appear
  - $a_n = a_{n-1} * a_{n-2}$ is non-linear (quadratic)
- homogenous = no additional terms
  - $a_n = a_{n-1} + n/2$ is non-homogenous because of the $n/2$ term
- constant coefficients = c_i s are constants, not functions of n
  - $a_n = n a_{n-1}$ does not have constant coefficients

Solving Recurrence Relations

So how to solve this recurrence relation?

$$a_n = c_1a_{n-1}+c_2a_{n-2}+\ldots+c_ka_{n-k}$$

Look for solutions of the form:

- $a_n = r^n$ for some constant r might work
- it works for $k = 1$
  - $a_n = ca_{n-1} = c(ca_{n-2}) = \ldots\ c^ia_{n-i} = c^na_0$

Let's see what that gives us:

$$r^n = c_1r^{n-1}+c_2r^{n-2}+\ldots+c_nr^{n-k}$$

Which can be rewritten as

$r^n - c_1r^{n-1}-c_2r^{n-2}-\ldots-c_nr^{n-k} = 0$ // divide by r^{n-k}

$r^k - c_1r^{k-1}-c_2r^{k-2}-\ldots-c_n = 0$

Called **characteristic equation** (also **characteristic polynomial**) of the recurrence relation

Solving Recurrence Relations

The roots of the characteristic equation are called **characteristic roots**

- every characteristic root satisfies the characteristic equation
- if the sequence $\{a_i\}$ satisfies the recurrence

  $a_n = c_1a_{n-1}+c_2a_{n-2}+\ldots+c_ka_{n-k}$,

  also the sequence $\{\alpha a_i\}$ satisfies it, for any constant α
  - corresponds to multiplying both sides by α
- actually, we can combine the solutions in a more complicated way
  - but let's do it only for $k=2$
    - we don't really know how to find characteristic roots for $k>2$
    - the case $k=1$ leads to simple geometric sequences, we know that

Solving Recurrence Relations

So, we have a recurrence $a_n = c_1 a_{n-1} + c_2 a_{n-2}$

The characteristic equation is

- $r^2 - c_1 r - c_2 = 0$
- there are two possibilities
  - two different roots r_1 and r_2
    - might be complex, shouldn't detract us too much
  - both roots are equal to each other

Solving Recurrence Relations

Consider first the case of two roots r_1 and r_2:

Theorem: The sequence $\{a_n\}$ is a solution to this recurrence relation if and only if $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$ for $n=0,1,2,\ldots$ where α_1 and α_2 are constants.

- if r_1 and r_2 are roots $\rightarrow$ $\{a_n\}$ is a solution for any constants α_1 and α_2

  using $r_1^2=c_1 r_1+c_2$ and $r_2^2=c_2 r^2+c_2$

- there are constants α_1 and α_2 such that $\{a_n\}$ satisfies the initial conditions for a_0 and a_1

- for fixed a_0 and a_1, the solution is unique

Solving Recurrence Relations - Examples

1. Consider $a_n = a_{n-1}+2a_{n-2}$, $a_0=2$, $a_1=7$
   - characteristic equation?
   - the roots?
   - α_1 and α_2?

2. Fibonacci numbers $f_n = f_{n-1}+f_{n-2}$, $f_1 = f_2 = 1$.

Solving Recurrence Relations

OK, but what if both roots are equal?

- characteristic equation is $r^2 - c_1 r - c_2 = (r - r_0)^2 = 0$ for some r_0
- $\alpha_1 r_0^n$ is still a solution, but it does not represent all possible solutions
  - it might not be enough to satisfy both a_0 and a_1

Theorem: Let $r^2 - c_1 r - c_2 = 0$ has one double solution r_0. A sequence $\{a_n\}$ is a solution of the recurrence relation $a_n = c_1 a_{n-1} + c_2 a_{n-2}$ if an only if $a_n = \alpha_1 r_0^n + \alpha_2 n r_0^n$, where α_1 and α_2 are constants.

Example: What is the solution for the recurrence relation $a_n = 6a_{n-1} - 9a_{n-2}$, with $a_0 = 1, a_1 = 6$?

Solving Recurrence Relations – k>2

Hm, what about the case $k>2$?

Analogous theorem holds:

Let $c_1, c_2, \ldots c_k$ be real numbers and the characteristic equation $r^k - c_1 r^{k-1} - \ldots - c_k = 0$ has k distinct roots $r_1, r_2, \ldots r_k$. Then a sequence $\{a_n\}$ is a solution of the recurrence relation $a_n = c_1 a_{n-1} + c_2 a_{n-2} \ldots c_k a_{n-k}$ if and only if $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n + \ldots + \alpha_k r_k^n$ for $n=0,1,2\ldots$, where $\alpha_1, \alpha_2, \ldots \alpha_k$ are constants.

OK, we are given a recurrence relation of order k

- can we find the characteristic equation?
  - easily
- can we find the roots?
  - now, this is tough, but we might get lucky and be able to factorize
- can we find $\alpha_1, \alpha_2, \ldots \alpha_k$?
  - tedious but straightforward solving of linear equalities

Solving Recurrence Relations – k>2

What about the case of multiple roots?

- analogous theorem holds (see Theorem 4 on p. 466)
- don’t need to remember exact details, but know that it exists and once you have the roots, you can solve the recurrency, even if the roots are not all distinct

More exercises

How many ways are there to cover **2xn** checkerboard using **1x2** and **2x2** tiles?

Find the solution for

- $a_n = 4a_{n-1} - 4a_{n-2}$ for $n>1$, with $a_0=6$, $a_1=8$
- $a_n = 7a_{n-1} - 10a_{n-2}$ for $n>1$, $a_0=2$, $a_1=1$
- $a_n = 7a_{n-2} + 6a_{n-3}$ with $a_0=9$, $a_1=10$ and $a_2=32$

Non-homogenous Recurrences

What about non-homogenous recurrences of the following form:

$a_n = c_1 a_{n-1} + c_2 a_{n-2} \ldots c_k a_{n-k} + F(n)$ for $n=0,1,2\ldots$,

where $c_1, c_2, \ldots c_k$ are constants?

Imagine that we have two solutions $\{a_n\}$ and $\{b_n\}$

Then $\{a_n - b_n\}$ is a solution to the homogenous recurrence relation

Theorem: If $\{a_n^p\}$ is a particular solution to a non-homogenous recurrence relation $a_n = a_n = c_1 a_{n-1} + c_2 a_{n-2} \ldots c_k a_{n-k} + F(n)$, then every solution is of the form $\{a_n^p + a_n^h\}$, where $\{a_n^h\}$ is a solution of the associated homogenous recurrence relation.

Non-homogenous Recurrences

We know how to solve homogenous recurrence relation

If we find one solution to the non-homogenous one, we can find all of them

But how to find that first solution?

- difficult, in general
- but we can do it when **F(n)** is good
  - product of a polynomial and s^n for a constant **s**
  - for example $F(n)= (n^2+5)3^n$

Non-homogenous Recurrences

Theorem: Suppose $\{a_n\}$ satisfies the linear non-homogenous recurrence relation $a_n = a_n = c_1 a_{n-1} + c_2 a_{n-2} \ldots c_k a_{n-k} + F(n)$, where $c_1, c_2, \ldots, c_k$ are real numbers and $F(n) = (b_t n^t + b_{t-1} n^{t-1} + \ldots + b_1 n + b_0) s^n$, where $b_1, b_2, \ldots b_t$ and s are real numbers.

When s is not a root of the characteristic equation of the associated homogenous recurrence relation, there is a particular solution of the form

$(p_t n^t + p_{t-1} n^{t-1} + \ldots p_1 n + p_0) s^n$.

When s is a root of this characteristic equation of multiplicity m, there is a particular solution of the form

$n^m (p_t n^t + p_{t-1} n^{t-1} + \ldots p_1 n + p_0) s^n$.

Exercises

Consider recurrence relation $a_n = 3a_{n-1}+2n$ with $a_1 = 3$.

The homogenous relation is $a_n = 3a_{n-1}$, and its solutions are $a_n = \alpha 3^n$ where α is a constant.

The characteristic equation is $r-3 = 0$, with a root of 3. In our case, $s= 1$, i.e. different from the root.

By the theorem, we are looking for a solution of the form $(cn+d)1^n = cn+d$

So, substitute it into the recurrence relation:

$$cn+d = 3(c(n-1)+d)+2n$$

$$cn+d = 3cn+2n-3c+3d$$

$$3c-2d = n(2c+2)$$

this must hold for every n, therefore $3c-2d = 0$ and $2c+2 = 0$, i.e. $c = -1$ and $d = -3/2$ and all solutions are of form $a_n = \alpha 3^n + (-n-3/2)$

To get $a_1 = 3$, we set $3 = a_1 = \alpha 3^1 + (-1-3/2)$, and $\alpha = 11/6$

More Exercises

Consider recurrence relation $a_n = 6a_{n-1} – 9a_{n-2} + F(n)$ with for $F(n)=$

- 3^n
- $n3^n$
- $n2^n$
- $(n^2+1)3^n$

What form does a particular solution have for each choice of $F(n)$?

Find a particular solution for $F(n) = n2^n$

- at least start

How to continue if we want a solution for $a_1 = 2, a_2 = 12$?

Divide and Conquer & Recurrences

Consider binary search algorithm. Let **BS(n)** be the number of comparison to perform the binary search of n elements. Then

BS(n) = BS(n/2)+2

Consider recursively finding maximum:

```
max(A[0..n-1] = max(max(A[0..n/2-1]), max(A[n/2..n-1])
```

M(n) = 2M(n/2)+2

Merge Sort:

```
Merge( MergeSort(A[0..n/2-1), MergeSort(A[n/2..n-1]))
```

- the cost of merging two sequences of **n/2** is at most **n**
- **MS(n) = 2MS(n/2)+n**

Divide and Conquer & Recurrences

Fast multiplication of **2n**-bit integers:

$$x = 2^n A_1 + A_0,\ y = 2^n B_1 + B_0$$

$$xy = (2^{2n} + 2^n) A_1 B_1 + 2^n (A_1 - A_0)(B_1 - B_0) + (2^n + 1) A_0 B_0$$

Total number of bit operations:

$$FM(2n) = 3FM(n) + Cn$$

Stassen Matrix Multiplication algorithm

- similar – divide each **nxn** matrix into **4 n/2 x n/2** matrices
- obtain the result as a sum of products of submatrices
- **7** matrix multiplications and **15** additions are need (of size **n/2 x n/2**)
- $S(n) = 7S(n/2) + 15(n/2)^2$

Divide and Conquer & Recurrences

General form:

$f(n) = af(n/b)+g(n)$

- but how to solve them?
- they are really not of the standard form we know so far
- we use **$f(n/2)$**, or, in general, **$f(n/b)$** instead of **$f(n-1), f(n-2)\ldots f(n-k)$**

Let's try expanding the general form to get some insight…

We get **$f(n) = a^k f(1) + \sum_{j=0}^{k-1} a^j g(n/b^j)$** where **$k = \log_b n$**

The result depends on the relationship of **a** and **b** and on **g(n)**

Divide and Conquer & Recurrences

First, simple case of **g(n)** being a constant **c**:

- the second term $\sum_{j=0}^{k-1} a^j g(n/b^j) = c\sum_{j=0}^{k-1} a^j$ is a geometric progression
- if $a = 1$, we get $O(ck)$ with k $= \log b\ n \in O(\log n)$ and $f(n) \in O(\log n)$
- if $a > 1$ we get the sum of diverging geometric progression
  - $f(n) = a^k f(1) + c(a^k - 1)/(a-1) = a^k(f(1) - c/(a-1)) - c(a-1) =$

  $= C_1 n^{\log_b a} + C_2$

Divide and Conquer & Recurrences

Applications for the case **g(n)** is constant:

- Consider binary search algorithm. Let **BS(n)** be the number of comparison to perform the binary search of n elements. Then

  BS(n) = BS(n/2)+2

- **b = 2, a = 1**, we get **BS(n) = O(log n)**

- Consider recursively finding maximum:

```
max(A[0..n-1] = max(max(A[0..n/2-1]), max(A[n/2..n-1])
```

  M(n) = 2M(n/2)+2

- **b = 2, a = 2,** we get **M(n) = $O(n^{\log_2 2}) = O(n)$**

Divide and Conquer & Recurrences

What about more general **g(n)**?

Master Theorem: Let **f** be an increasing function that satisfies

$$f(n) = af(n/b)+cn^d$$

Whevever $n = b^k$, where **k** is a positive integer, $a \geq 1$, **b** is integer greater then **1** and **c** and **d** are real numbers with **c** positive and **d** nonnegative. Then

$O(n^d)$ if $a<b^d$

$f(n)$ is $O(n^d \log n)$ if $a=b^d$

$O(n^{\log_b a})$ if $a>b^d$

Applications:

- merge sort, quasi-parallel merge sort, fast integer multiplication, Strassen's algorithm
- closest pair problem