MAT2540, Classwork11, Spring2026

8.2 Solving Linear Recurrence Relations (Conti.)

17. How to guess the form of the particular solution $a_n^{(p)}$ of a linear non-homogeneous recurrence relation

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k} + F(n)?$$

The form of the particular solution $a_n^{(p)}$ depends on the function $F(n)$:

Non-homogeneous term $F(n)$	Trial solution for $a_n^{(p)}$
Polynomial	
$F(n) = a$ where a is a constant	$a_n^{(p)} =$ A where A is a constant
$F(n) = n$	$a_n^{(p)} =$ $An+B$ where A, B are constants
$F(n) = an + b$ where a, b are constants	$a_n^{(p)} =$ $An+B$ where A, B are constants
$F(n) = an^2 + bn + c$ where a, b, c are constants	$a_n^{(p)} =$ An^2+Bn+C where A, B, C are constants
Power/exponential functions	
$F(n) = 2^n$	$a_n^{(p)} =$ $A\cdot 2^n$ where A is a constant
$F(n) = -5^n$	$a_n^{(p)} =$ $A\cdot 5^n$ where A is a constant
$F(n) = e^n$	$a_n^{(p)} =$ $A\cdot e^n$ where A is a constant
Combination functions	
$F(n) = n + 2^n$	$a_n^{(p)} =$ $An+B+C\cdot 2^n$ where A, B, C are constants
$F(n) = (n + a)2^n$	$a_n^{(p)} =$ $(An+B)\cdot 2^n$ where A, B are constants

Here, a, b, c are known constants and A, B, C are undetermined constants and can be found by using initial data

18.In general, we need to consider the characteristic roots r_0 of associated homogeneous recurrence relation:

Is $s = r_0$?	Non-homogeneous term $F(n)$	Trial solution for $a_n^{(p)}$
No	$F(n) = (b_t n^t + b_{t-1} n^{t-1} + \cdots + b_1 n^1 + b_0)s^n$	$a_n^{(p)} =$ $(P_t n^t + P_{t-1} n^{t-1} + P_{t-2} n^{t-2} + \cdots + P_2 n^2 + P_1 n + P_0)s^n$
Yes	$F(n) = (b_t n^t + b_{t-1} n^{t-1} + \cdots + b_1 n^1 + b_0)s^n$	$a_n^{(p)} =$ $n^m(P_t n^t + P_{t-1} n^{t-1} + P_{t-2} n^{t-2} + \cdots + P_2 n^2 + P_1 n + P_0)s^n$

Here m is the multiplicity of r_0.

Let $a_n^{(h)} = r^n$ → $a_n = 6a_{n-1} - 9a_{n-2} \Rightarrow r^n = 6r^{n-1} - 9r^{n-2}$

19. Find the trial solution $a_n^{(p)}$ for $a_n = 6a_{n-1} - 9a_{n-2} + F(n)$ when (a) $F(n) = 2^n$, (b) $F(n) = (n^2 + 1)3^n$, (c) $F(n) = 3^n$, (d) $F(n) = n3^n$, (e) $F(n) = n^2 2^n$.

$r^2 - 6r + 9 = 0$

$(r-3)^2 = 0$

$r = 3$

$a_n^{(h)} = \alpha_1 3^n + \alpha_2 n 3^n$

(a) $a_n^{(p)} = A2^n$

(b) $a_n^{(p)} = n^2(An^2+Bn+C)3^n$

(c) $a_n^{(p)} = An3^n$

(d) $n^2(An+B)\cdot 3^n$

(e) $a_n^{(p)} (An^2+Bn+C)2^n$

20. Find the solution of $a_n = 6a_{n-1} - 9a_{n-2} + 2^n$ with $a_1 = 11$ and $a_2 = 7$.

① Find $a_n^{(h)}$

Let $a_n^{(h)} = r^n$,

$r^n = 6r^{n-1} - 9r^{n-2}$

$\Rightarrow r^{n-2}(r^2 - 6r - 9) = 0$

$\Rightarrow (r-3)^2 = 0 \Rightarrow r = 3$

$a_n^{(h)} = \alpha_1(3)^n + \alpha_2 n(3)^n$

② Find $a_n^{(p)}$

Let $a_n^{(p)} = A2^n$, $a_{n-1}^{(p)} = A2^{n-1}$, $a_{n-2}^{(p)} = A2^{n-2}$

$A2^n = 6\cdot A2^{n-1} - 9A2^{n-2} + 2^n$

$4A = 12A - 9A + 4$

$A = 4$, $a_n^{(p)} = 4\cdot(2)^n$

③ Find α_1, α_2

$a_n = \alpha_1(3)^n + \alpha_2 n(3)^n + 4(2)^n$

$11 = a_1 = (\alpha_1 + \alpha_2)(3)^1 + 4\cdot(2)^1$

$7 = a_2 = (\alpha_1 + 2\alpha_2)(3)^2 + 4\cdot(2)^2$

$\Rightarrow 11 = 3(\alpha_1 + \alpha_2) + 8$

$7 = 9(\alpha_1 + 2\alpha_2) + 16$

$\Rightarrow \alpha_1 + \alpha_2 = 1$, $\alpha_1 + 2\alpha_2 = -1$ $\Rightarrow \alpha_2 = -2$, $\alpha_1 = 3$

$\Rightarrow a_n = 3(3)^n - 2n(3)^n + 4(2)^n$

19. $a_n = 6a_{n-1} - 9a_{n-2} + 3^n$ particular solution

$a_n^{(h)} = \alpha_1 (3)^n + \alpha_2 n (3)^n$, $a_n^{(p)} = n^2 A 3^n$

For $a_n^{(p)}$, we have

$a_n^{(p)} = 6a_{n-1}^{(p)} - 9a_{n-2}^{(p)} + 3^n$

$An^2 3^n = 6 \cdot A(n-1)^2 3^{n-1} - 9 \cdot A(n-2)^2 3^{n-2} + 3^n$

$3^n \neq 0$ $An^2 3^n = 2A(n-1)^2 3^n - A(n-2)^2 \cdot 3^n + 3^n$

$= 3^n \left(2A(n-1)^2 - A(n-2)^2 + 1 \right)$

$An^2 = 2A(n^2 - 2n + 1) - A(n^2 - 4n + 4) + 1$

$An^2 = (2A - A)n^2 + (-4A + 4A)n + (2A - 4A + 1)$

$An^2 = An^2 + (-2A + 1)$

$0 = An^2 + (-2A + 1) - An^2 \Rightarrow -2A + 1 = 0 \Rightarrow A = \frac{1}{2}$

8.3 Divide-and-Conquer Algorithms and Recurrence Relations

1. Definition: Divide-and-Conquer Algorithms.

An algorithm is called divide-and-conquer algorithm if it divides a problem into one or more instances of the same problem of smaller size and they conquer the problem by using the solutions of the smaller problems to find a solution of the original problem.

2. Definition: Divide-and-Conquer Recurrence Relations.

If $f(n)$ represents the number of operations required to solve the problem of size n, it follows that f satisfies the recurrence relation

$$f(n) = \alpha f\left(\frac{n}{\beta}\right) + g(n)$$

This is called a divide-and-conquer recurrence relation which is a recursive algorithm dividing a problem of size n into α subproblems, where each subproblem is of size n/β and total of $g(n)$ extra operations which are required in the conquer step of the algorithm to combine the solutions of the subproblems into a solution of the original problem.

3. Example: Binary Search.

The problem of size n has been reduced to $\alpha =$ one problem of size $n/2$ and two comparisons are needed: $\beta = 2$

1) $i < j$: check if any term left in the list
2) $x > a_m$: decide which half to keep

The divide-and-conquer recurrence relation:

$f(n) = 1 \cdot f\left(\frac{n}{2}\right) + 2$

ALGORITHM 3 The Binary Search Algorithm.

procedure *binary search* (x: integer, $a_1, a_2, \ldots, a_n$: increasing integers)
$i := 1$ {i is left endpoint of search interval}
$j := n$ {j is right endpoint of search interval}
while $i < j$
  $m := \lfloor (i + j)/2 \rfloor$
  if $x > a_m$ **then** $i := m + 1$
  else $j := m$
if $x = a_i$ **then** *location* $:= i$
else *location* $:= 0$
return *location* {*location* is the subscript i of the term a_i equal to x, or 0 if x is not found}

4. Example: Merge Sort.

This algorithm splits a list to be sorted with n items into $\alpha =$ two lists with $\frac{n}{2} \leq \beta$ elements each, and uses fewer than n comparisons to merge the two sorted lists of $n/2$ items each into one sorted list.

The divide-and-conquer recurrence relation: $M(n) = 2M\left(\frac{n}{2}\right) + n$ ← comparisons when merging

ALGORITHM 9 A Recursive Merge Sort.

procedure *mergesort*($L = a_1, \ldots, a_n$)
if $n > 1$ **then**
  $m := \lfloor n/2 \rfloor$
  $L_1 := a_1, a_2, \ldots, a_m$
  $L_2 := a_{m+1}, a_{m+2}, \ldots, a_n$
  $L := merge(mergesort(L_1), mergesort(L_2))$
{L is now sorted into elements in nondecreasing order}

5. Given a recurrence relation of the form $f(n) = \alpha f\left(\frac{n}{\beta}\right) + g(n)$. Derive estimates of the size of functions that satisfy such recurrence relations.

$f\left(\frac{n}{\beta}\right) = \alpha f\left(\frac{n}{\beta^2}\right) + g\left(\frac{n}{\beta}\right)$ $\quad f\left(\frac{n}{\beta^2}\right) = \alpha f\left(\frac{n}{\beta^3}\right) + g\left(\frac{n}{\beta^2}\right)$

Suppose that f satisfies this recurrence relation whenever n is divisible by β. Let $n = \beta^k$, where k is a positive integer.

Then $f(n) = \alpha f\left(\frac{n}{\beta}\right) + g(n) = \alpha\left(\alpha f\left(\frac{n}{\beta^2}\right) + g\left(\frac{n}{\beta}\right)\right) + g(n) = \alpha^2 f\left(\frac{n}{\beta^2}\right) + \alpha g\left(\frac{n}{\beta}\right) + g(n)$

$= \alpha^2\left(\alpha f\left(\frac{n}{\beta^3}\right) + g\left(\frac{n}{\beta^2}\right)\right) + \alpha g\left(\frac{n}{\beta}\right) + g(n) = \alpha^3 f\left(\frac{n}{\beta^3}\right) + \alpha^2 g\left(\frac{n}{\beta^2}\right) + \alpha g\left(\frac{n}{\beta}\right) + g(n)$

$= \cdots = \alpha^k f\left(\frac{n}{\beta^k}\right) + \alpha^{k-1} g\left(\frac{n}{\beta^{k-1}}\right) + \alpha^{k-2} g\left(\frac{n}{\beta^{k-2}}\right) + \cdots + \alpha g\left(\frac{n}{\beta}\right) + g(n)$

$n = \beta^k$ $\quad = \alpha^k f(1) + \sum_{j=0}^{k-1} \alpha^j \cdot g\left(\frac{n}{\beta^j}\right)$

$f(n) = \quad f(1) + \_\_\_\_$