CS/MATH 111, Discrete Structures - Fall 2018. Discussion 6 - Non-homogeneous Recurrences, Divide and Conquer & Inclusion - Exclusion

Andres, Sara, Elena

University of California, Riverside

November 2, 2018

Outline

Non-homogeneous recurrence[1]

Theorem

$$f_n = f_n^{'} + f_n^{''}$$

If $\{f_n^{''}\}$ *is a particular solution of the non-homogeneous linear recurrence relation with constant coefficients:*

$$f_n = c_1 \cdot f_{n-1} + c_2 \cdot f_{n-2} + \cdots + c_k \cdot f_{n-k} + g(n)$$

then every solution is of the form $\{f_n^{'} + f_n^{''}\}$*, where* $\{f_n^{'}\}$ *is a solution of the associated homogeneous recurrence relation.*

[1] Proof available at [Rosen, 2015. pg 521].

Non-homogeneous recurrence

Linear Non-Homogeneous Recurrence Relations

$f_n = 6f_{n-1} - 9f_{n-2} + g(n)$

$f'_{nc} = \alpha_1 3^n + \alpha_2 n 3^n$

$f''_n = n^m(p_t n^t + p_{t-1} n^{t-1} + \dots + p_1 n + p_0)\, s^n$

$g(n) = 5$	$f''_n = p_0$
$g(n) = 5n + 1$	$f''_n = p_1 n + p_0$
$g(n) = 5n^2 + 1$	$f''_n = p_2 n^2 + p_1 n + p_0$
$g(n) = 5n^2 + n + 1$	$f''_n = p_2 n^2 + p_1 n + p_0$
$g(n) = n2^n$	$f''_n = (p_1 n + p_0)\, 2^n$
$g(n) = 2^n(5n^2 + n + 1)$	$f''_n = (p_2 n^2 + p_1 n + p_0)\, 2^n$

Non-homogeneous recurrence

- Find a particular solution for recurrence relation:

$$f_n = 3 \cdot f_{n-1} + f_{n-2} + 6 \tag{1}$$

Non-homogeneous recurrence

- Find a particular solution for recurrence relation:

$$f_n = 3 \cdot f_{n-1} + f_{n-2} + 6 \tag{1}$$

- $g(n) = 6$, so:

$$f_n^{''} = p_0 \tag{2}$$

Non-homogeneous recurrence

- Find a particular solution for recurrence relation:

$$f_n = 3 \cdot f_{n-1} + f_{n-2} + 6 \tag{1}$$

- $g(n) = 6$, so:

$$f_n^{''} = p_0 \tag{2}$$

- Plugin (2) in (1) becomes:

$$p_0 = 3 \cdot p_0 + p_0 + 6$$

$$p_0 - p_0 - 3 \cdot p_0 = 6$$

$$p_0 = -\frac{6}{2} = -2 \tag{3}$$

Non-homogeneous recurrence

- Find a particular solution for recurrence relation:

$$f_n = 3 \cdot f_{n-1} + f_{n-2} + 6 \tag{1}$$

- $g(n) = 6$, so:

$$f_n^{''} = p_0 \tag{2}$$

- Plugin (2) in (1) becomes:

$$p_0 = 3 \cdot p_0 + p_0 + 6$$

$$p_0 - p_0 - 3 \cdot p_0 = 6$$

$$p_0 = -\frac{6}{2} = -2 \tag{3}$$

- Finally, (3) in (2):

$$f_n^{''} = -2$$

Non-homogeneous recurrence

- Find a particular solution for recurrence relation:

$$f_n = 3 \cdot f_{n-1} + f_{n-2} + 3 \cdot 2^n \tag{1}$$

Non-homogeneous recurrence

- Find a particular solution for recurrence relation:

$$f_n = 3 \cdot f_{n-1} + f_{n-2} + 3 \cdot 2^n \tag{1}$$

- $g(n) = 3 \cdot 2^n$, so:

$$f_n^{''} = p_0 \cdot 2^n \tag{2}$$

Non-homogeneous recurrence

- Plug (2) in (1) becomes:

$$
\begin{aligned}
p_0 \cdot 2^n &= 3 \cdot p_0 \cdot 2^{n-1} + p_0 \cdot 2^{n-2} + 3 \cdot 2^n \\
p_0 \cdot 2^n &= 2^{n-2}(3 \cdot p_0 \cdot 2^1 + p_0 \cdot 2^0 + 3 \cdot 2^2) \\
p_0 \cdot 2^2 &= 3 \cdot p_0 \cdot 2 + p_0 + 3 \cdot 4 \\
p_0 \cdot 4 &= 7 \cdot p_0 + 12 \\
p_0 &= -4 \qquad (3)
\end{aligned}
$$

Non-homogeneous recurrence

- Plug (2) in (1) becomes:

$$p_0 \cdot 2^n = 3 \cdot p_0 \cdot 2^{n-1} + p_0 \cdot 2^{n-2} + 3 \cdot 2^n$$
$$p_0 \cdot 2^n = 2^{n-2}(3 \cdot p_0 \cdot 2^1 + p_0 \cdot 2^0 + 3 \cdot 2^2)$$
$$p_0 \cdot 2^2 = 3 \cdot p_0 \cdot 2 + p_0 + 3 \cdot 4$$
$$p_0 \cdot 4 = 7 \cdot p_0 + 12$$
$$p_0 = -4 \tag{3}$$

- Finally, (3) in (2):

$$f_n^{''} = -4 \cdot 2^n$$

Non-homogeneous recurrence

Solve the following non-homogeneous recurrence:

$$f_n = 4 \cdot f_{n-1} - 4 \cdot f_{n-2} + 2 \cdot 5^n \tag{1}$$

with initial condition: $f_0 = 1$ and $f_1 = 2$.

Non-homogeneous recurrence

Solve the following non-homogeneous recurrence:

$$f_n = 4 \cdot f_{n-1} - 4 \cdot f_{n-2} + 2 \cdot 5^n \tag{1}$$

with initial condition: $f_0 = 1$ and $f_1 = 2$.

- $f_n^{'} = 4 \cdot f_{n-1} - 4 \cdot f_{n-2}$
  1. Caractheristic equations and its roots:

  $$x^2 - 4 \cdot x + 4 = 0$$

  $$(x-2)(x-2) = 0$$

  $$x_{1,2} = 2$$

  2. General form of the solution:

  $$f_n^{'} = \alpha_1 \cdot 2^n + \alpha_2 \cdot n \cdot 2^n$$

Non-homogeneous recurrence

Solve the following non-homogeneous recurrence:

$$f_n = 4 \cdot f_{n-1} - 4 \cdot f_{n-2} + 2 \cdot 5^n \quad (1)$$

with initial condition: $f_0 = 1$ and $f_1 = 2$.

- $g(n) = 2 \cdot 5^n$, so:

$$f_n^{''} = p_0 \cdot 5^n \quad (2)$$

- Plug (2) in (1) becomes:

$$p_0 \cdot 5^n = 4 \cdot p_0 \cdot 5^{n-1} - 4 \cdot p_0 \cdot 5^{n-2} + 2 \cdot 5^n$$

$$p_0 = \frac{50}{9} \quad (3)$$

- Finally, (3) in (2):

$$f_n^{''} = \frac{50}{9} \cdot 5^n$$

Non-homogeneous recurrence

Solve the following non-homogeneous recurrence:

$$f_n = 4 \cdot f_{n-1} - 4 \cdot f_{n-2} + 2 \cdot 5^n \tag{1}$$

with initial condition: $f_0 = 1$ and $f_1 = 2$.

- $f_n' = \alpha_1 \cdot 2^n + \alpha_2 \cdot n \cdot 2^n$
- $f_n'' = \frac{50}{9} \cdot 5^n$
- $f_n = \alpha_1 \cdot 2^n + \alpha_2 \cdot n \cdot 2^n + \frac{50}{9} \cdot 5^n$

3 Initial condition equations and their solutions:

$$\begin{aligned} f_0 = \quad & \alpha_1 \cdot 2^0 + \alpha_2 \cdot 0 \cdot 2^0 + \tfrac{50}{9} \cdot 5^0 \quad && = \alpha_1 + \tfrac{50}{9} = 1 \\ f_1 = \quad & \alpha_1 \cdot 2^1 + \alpha_2 \cdot 1 \cdot 2^1 + \tfrac{50}{9} \cdot 5^1 \quad && = 2 \cdot \alpha_1 + 2 \cdot \alpha_2 + 5 \cdot \tfrac{50}{9} = 2 \end{aligned}$$

where $\alpha_1 = -\frac{41}{9}$ and $\alpha_2 = -\frac{25}{3}$.

4 Final answer:

$$\vdots$$

Outline

Divide and Conquer

Theorem

Let $a \geq 0$, $b > 0$, $c > 0$ and $d \geq 0$. If $T(n)$ satisfies the recurrence then

$$T(n) = a \cdot T\left(\frac{n}{b}\right) + c \cdot n^d$$

$$T(n) = \begin{cases} \Theta(n^{\log_b a}) & a > b^d \\ \Theta(n^d \log n) & a = b^d \\ \Theta(n^d) & a < b^d \end{cases}$$

Divide and Conquer

- Give the asymptotic value (using the Θ-notation) for the number of letters that will be printed by the following algorithms. You need to provide an appropriate recurrence equation and its solution.

Divide and Conquer

```
def PrintXs(n: integer)
    if(n < 3)
        print("X")
    else
        PrintXs(n/3)
        PrintXs(n/3)
        PrintXs(n/3)
        for(i <- 1 to 2*n)
            print("X")
```

Divide and Conquer

```
def PrintXs(n: integer)
    if(n < 3)
        print("X")
    else
        PrintXs(n/3)
        PrintXs(n/3)
        PrintXs(n/3)
        for(i <- 1 to 2*n)
            print("X")
```

- We have 3 recursive calls, each with parameter $\frac{n}{3}$.
- Recurrence is $X(n) = 3 \cdot X\left(\frac{n}{3}\right) + 2 \cdot n$.
- If $a = 3$, $b = 3$, $c = 2$ and $d = 1$ then $a = b^d$ and $\Theta(n^d \log n)$.

Divide and Conquer

```
def PrintYs(n: integer)
    if(n < 2)
        print("Y")
    else
        for(i <- 1 to 16)
            PrintYs(n/2)
        for(i <- 1 to n^3)
            print("Y")
```

Divide and Conquer

```
def PrintYs(n: integer)
    if(n < 2)
        print("Y")
    else
        for(i <- 1 to 16)
            PrintYs(n/2)
        for(i <- 1 to n^3)
            print("Y")
```

- We have 16 recursive calls, each with parameter $\frac{n}{2}$.
- Recurrence is $X(n) = 16 \cdot X\left(\frac{n}{2}\right) + n^3$.
- If $a = 16$, $b = 2$, $c = 1$ and $d = 3$ then $a > b^d$ and $\Theta(n^{\log_2 16})$.

Divide and Conquer

```
def PrintZs(n: integer)
    if(n < 3)
        print("Z")
    else
        PrintZs(n/3)
        PrintZs(n/3)
        for(i <- 1 to 7*n)
            print("Z")
```

Divide and Conquer

```
def PrintZs(n: integer)
    if(n < 3)
        print("Z")
    else
        PrintZs(n/3)
        PrintZs(n/3)
        for(i <- 1 to 7*n)
            print("Z")
```

- We have 2 recursive calls, each with parameter $\frac{n}{3}$.
- Recurrence is $X(n) = 2 \cdot X\left(\frac{n}{3}\right) + 7 \cdot n$.
- If $a = 2$, $b = 3$, $c = 7$ and $d = 1$ then $a < b^d$ and $\Theta(n)$.

Divide and Conquer

(d) **Algorithm** PrintUs (n : **integer**)

```
if n < 4
    print("U")
else
    PrintUs(⌈n/4⌉)
    PrintUs(⌊n/4⌋)
    for i ← 1 to 11 do print("U")
```

(e) **Algorithm** PrintVs (n : **integer**)

```
if n < 3
    print("V")
else
    for j ← 1 to 9 do PrintVs(⌊n/3⌋)
    for i ← 1 to 2n^3 do print("V")
```

Divide and Conquer

(d)
There are 2 recursive calls, each with parameter $\lceil n/4 \rceil$. Since we are looking for an asymptotic solution, we can ignore rounding. Then the number of letters printed can be expressed by the recurrence:

$$X(n) = 2X(n/4) + 11.$$

We apply the Master Theorem with $a = 2$, $b = 4$, $c = 11$, $d = 0$. Here, we have $a > b^d$, so the solution is $\Theta(n^{\log_4 2})$.

(e)
There are 9 recursive calls, each with parameter $\lfloor n/3 \rfloor$. Since we are looking for an asymptotic solution, we can ignore rounding. Then the number of letters printed can be expressed by the recurrence:

$$X(n) = 9X(n/3) + 2n^3.$$

We apply the Master Theorem with $a = 9$, $b = 3$, $c = 2$, $d = 3$. Here, we have $a < b^d$, so the solution is $\Theta(n^3)$.

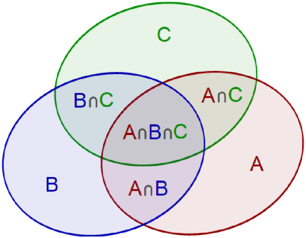
Outline

Inclusion-Exclusion

Problem 2: We have a group of people, each of which is a citizen of either US or Mexico or Canada. Half of the people in this group are US citizens, 10 are Mexican citizens, 17 are Canadian citizens, 4 people have dual US-Mexican citizenship, 5 have US-Canadian citizenship, 6 have Canadian-Mexican, and 2 are citizens of all three countries. How many people are in this group? Show your work.

Inclusion-Exclusion

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$$



Inclusion-Exclusion

$$\begin{aligned}
\text{Us citizens:} & \quad \mid A \mid & = \tfrac{X}{2} \\
\text{Mexican citizen:} & \quad \mid B \mid & = 10 \\
\text{Canadian citizen:} & \quad \mid C \mid & = 17 \\
\text{US-Mexican citizen:} & \quad \mid A \cap B \mid & = 4 \\
\text{US-Canadian citizen:} & \quad \mid A \cap C \mid & = 5 \\
\text{Canadian-Mexican:} & \quad \mid B \cap C \mid & = 6 \\
\text{Citizens of all countries:} & \quad \mid A \cap B \cap C \mid & = 2
\end{aligned}$$

$$X = \frac{X}{2} + 10 + 17 - 4 - 5 - 6 + 2$$

$$X = \frac{X}{2} + 14$$

$$X = 28$$