## Cormen Chapter 4 Recurrences

(1) The SUBSTITUTION METHOD
(2) Iteration Method.
(3) Master Method.

Substitution ... Assumes a good guess for the complexity exists.
consider,
T(n) = 2T ($\lfloor n/2 \rfloor$) + n
which is an approximate recurrence for Merge sort.

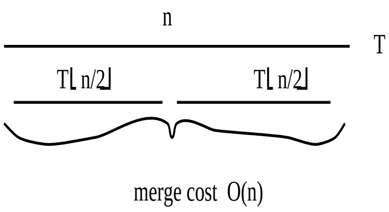

Figure 1:

We guess the solution to be O(n log n)
THIS IMPLIES the complexity is less than: $c(nlogn)$1 We try to confirm this by substitution.

$$T(n) \leq 2T(\lfloor n/2 \rfloor) + n \qquad\qquad T(1) = 1$$

$$T(n) \leq 2(c.\lfloor n/2 \rfloor.log(\lfloor n/2 \rfloor)) + n$$

$\leq$ c.n.lg($\lfloor n/2 \rfloor$)+n
$\leq$ c.n.lg n - c.n.lg 2 + n
$\leq$ c.n.lg n - c.n + n
$\leq$ c.n.lg n
which holds for c$\geq$ 1

**Slide Lecture 1 –3**

Now consider Boundary conditions for $n_0$
→ does work for T(1) since c1log1 = 0
T(2)=4    T(3) = 5
are derived from the recurrence relation
T(3) = 2T(1) + 3 = 5
T(2) = 2T(1) + 2 = 4
So T(2) $\leq$ c(2 lg 2) and T(3) $\leq$ c(3 lg 3)
for any c $\geq$ 2

**Slide Lecture 1 –4**

Substitute : correcting a guess
T(n) = T($\lfloor n/2 \rfloor$) + $T(\lfloor n/2 \rfloor)$ + 1
Quess Order(n) and substitute n
T(n) $\leq c.\lfloor n/2 \rfloor + c(\lfloor n/2 \rfloor)$ + 1
$\leq c.n$ + 1 NO!

Induction doesn't work *unless we prove the exact form.* The result must be LESS than the actual bound.

But we are close. Use cn-b

$$T(n) \leq (c.\frac{n}{\lfloor 2 \rfloor - b}) + (c\frac{n}{\lfloor 2 \rfloor - b}) + 1$$
$$= c.n - 2b + 1$$

which is $\leq cn - b$ for b $\geq$ 1

2. ITERATION

Consider $T(n) = 3T(\lfloor n/4 \rfloor) + n$
Place recursion on the far right & iterate :
$T(n) = n + 3T(\lfloor n/4 \rfloor)$
$= n + 3(\lfloor n/4 \rfloor) + 3T(n/4\lfloor/4\rfloor))$
$= n + 3(\lfloor n/4 \rfloor + 3(\lfloor n/16 \rfloor + 3T(n/64)))$
$= n + 3n/4 + 9n/16 + 27T(n/64)$

The $i^{th}$ term is $3^i$n/$4^i$
Since n is reduced by 4 at each iteration it continues for $log_4$N steps.

So

$T(n) \leq n + 3n/4 + 9n/16 + 27n/64 + \ldots + 3^{log_4 N}$ N / N

$$\leq n. \sum_{i=0}^{log_4 N-1} (3/4)^i + \mathcal{O}(3^{log_4 N})$$

$$\leq n. \sum_{i=0}^{\infty}(3/4)^i + \mathcal{O}(3^{log_4 N})$$

$\leq$ 4n + $\mathcal{O}$(n) $\quad$ $\leq$ 5n

= $\mathcal{O}$(n)

$3^{log_4 N}$ can be rewritten

$N^{log_4 3}$ which is less than N

**Slide Lecture 1 –7**

---

So,

$$\mathcal{O}(N^{log_4 3} = o(N) \quad \text{"little-o"}$$

$$\sum_{i=0}^{log_4 N} (3/4)^i < \sum_{i=0}^{\infty}(3/4)^i$$

1 + 3/4 + 4/16 + 27/16

Bounding a series

$$\sum_{k=0}^{\infty} r^k = \frac{1}{1-r}$$

(cormen pg 47)

so

**Slide Lecture 1 –8**

$$\sum_{k=0}^{\infty}(3/4)^k = \frac{1}{1-3/4} = \frac{1}{1/4} = 4$$

EXAMPLE 2.6.2. Find an explicit formula for $c_n$, the number of moves in which the n-disk Tower of Hanoi puzzle can be solved.

In Example 2.5.7, we obtained the recurrence relation

$c_n = 2c_{n-1} + 1$

and the initial condition

$c_1 = 1$

Applying the iterative method to (2.6.1), we obtain

$c_n = 2c_{n-1} + 1$

$= 2(2c_{n-2} + 1) + 1$

$= 2^2c_{n-2} + 2 + 1$

$= 2^2(2c_{n-3} + 1) + 2 + 1$

$= 2^3c_{n-3} + 2^2 + 2 + 1$

.

.

.

=$2^{n-1}c_1 + 2^{n-2} + 2^{n-3} + \ldots + 2 + 1$

=$2^{n-1} + 2^{n-2} + 2^{n-3} + \ldots + 2 + 1$

$$\sum_{i=0}^{n-1} 2^i = 2^N - 1$$

=$2^n$ - 1

**Slide Lecture 1 –11**

The last step results from the formula for the sum of a geometric series.

Note :

$$\sum_{i=0}^{n} x^i = \frac{x^{n+1} - 1}{x - 1}$$

so

$$\sum_{i=0}^{n-1} 2^i = \frac{2^n - 1}{1}$$

**Slide Lecture 1 –12**

Master Method

* A ”cookbook” for solving recurrence relations
Form T(n) = aT(n/b) + f(n)
$a \geq 1$ , b >1
f(n) asymptotically positive
Problem of size **N** is divided into **a** subproblems of size N/b.
Subproblems are solved recursively in time T(n/b). The cost of dividing and combining results is given by f(n)
Technically, n/b should be $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$
but this detail is ignored in practice.

Cormen
Master Method : 3 cases
T(n) = aT(n/b) + f(n)
(1) If f(n) = $\mathcal{O}(n^{log_b a - \in})$ for some
constant $\in > 0$, then

$$T(n) = \Theta(n^{log_b a})$$

**i.e.** $\mathbf{f(n) < n^{log_b a}}$

(2) If $f(n) = \Theta(n^{log_b a})$ then

$$T(n) = \Theta(n^{log_b a}\ lg\ n)$$
$$i.e. \quad f(n) = c(n^{log_b a})$$

(3) If f(n) = $\Omega(n^{log_b a + \in})$ for some

constant $\in > 0$ and if
a f(n/b) $\leq cf(n)$ for some
constant $c < 1$ and all sufficiently large n, then

$$T(n) = \mathcal{O}(f(n))$$

$$i.e. \quad f(n) > n^{log_b a}$$

**CASE 1**
f(n) is polynomially smaller than $n^{log_b a}$ by a factor of $n^{\in}$.
so $n^{log_b a}$ dominates.

**CASE 2**
f(n) $= n^{log_b a}$
so the cost at each step is $n^{log_b a}$ and there are lg n steps
Therefore, $\Theta(n^{log_b a} lgn)$

**CASE 3**
f(n) $> n^{log_b a}$ by a polynomial factor.
so the dominant term is
$f(n) + f(\frac{n}{b}) + f(\frac{n/b}{b})$ ....
$\mathcal{O}$(f(n))

T(n) = 9T(n/3) + n
so a=9, b=3 , f(n)=n
$n^{log_3 9} = n^2$
f(n) = $\mathcal{O}(n^{log_3 9-1})$ so *CASE 1*
T(n) = $\Theta(n^2)$

T(n) = T(2n/3) + 1
a=1 , b=3/2 , f(n)=1
$log_{3/2} 1 = 0$, so $n^{log_b a} = 1$.
f(n) = $n^{log_b a}$ so *CASE 2*
T(n) = 1.lg n

T(n) = 3T(n/4) + n lg n
a=3 , b=4 , f(n)=n lg n
$n^{log_b a} = n^{log_4 3} = \mathcal{O}(n^{0.793})$
f(n) = $\Omega(n^{log_4 3} + \in)$ because
$\Omega(n^{0.79+0.2}) = \Omega(n^{0.99})$
$= o(n)$
so f(n) dominates ... *CASE 3*

A NON-CASE ...
T(n) = 2T(n/2) + n lg n
a=2 , b=2 , $log_b a = 1$ , $n^{log_b a}$ = n
f(n) = n log n
n lg n is asymptotically larger than n but not polynomially larger .
$\frac{nlgn}{n}$ = lg n , lg n < $n^{\in}$

Master Method (But more General)

DIVIDE & CONQUER FORM OF REC. REL.

$a_n = ca_{n/d}$ + f(n) ...... (1)

Take f(n) = $\alpha n^p$

The following table indicates the form of the leading term of solutions for (1) for some common values of c,d,p ($d \geq 2$)

| | | |
|---|---|---|
| c=1 | p=0 | $a_n = \mathcal{O}(lgn)$ |
| c<d | p=1 | $a_n = \mathcal{O}(n)$ |
| c=d | p=0 | $a_n = \mathcal{O}(n)$ |
| c=d | p=1 | $a_n = \mathcal{O}(nlgn)$ |

**Remember :** Binary Search: $a_n = a_{n/2} + c \rightarrow$ log n
Merge Sort: $a_n = 2a_{n/2} + n \rightarrow nlogn$
MAX (Divco): $a_n = 2a_{n/2} + c \rightarrow$n

Multiply two n-digit numbers
count the digit × digit multiplcation

1) Iterative is $n \times n$ $O(n^2)$

```
1 2 3 4
5 6 7 8
---------
x x x x
```

2) Recursive DivCO

Split numbers into $n/2$ digit numbers
$g = g_1 \times 10^{n/2} + g_2 (12 \times 10^2 + 34)$
$h = h_1 \times 10^{n/2} + h_2 (56 \times 10^2 + 78)$

$$g \times h = g_1 h_1 10^n + g_1 h_2 10^{n/2} + g_2 h_1 10^{n/2} + g_2 h_2$$

$$g_1 h_2 + g_2 h_1 = (g_1 + g_2)(h_1 + h_2) - g_1 h_1 - g_2 h_2$$

So we need three $n/2$ digit multiplications

$g_1 h_1, g_2 h_2, (g_1 + g_2)(h_1 + h_2)$
$An = 3A(n/2)$
$An = O(n \log_2 3)$
$3A(n/2) nlog_2 3 = O(n^{1.6})$

## Time Analysis of Recursion

General form of Divide & Conquer :

```
DivCo(P) → R
    if trivial(P) then Solve_Direct(P) → R
    else split(P) → $P_1 \ldots P_n$
        for i in 1 ... n DivCo($P_i$) → $r_i$
        combine($r_1 ... r_n$) → $R$
```

The complextiy of DivCo depends on
- Complexity of trivial, Solve_direct, split, combine.
- The number and size of subprograms $P_1 ... P_n$

Expressing this complexity leads to a *Recurrence Relation :*

**f(n) = h(f(1), f(2) , ... , f(n-1))**

NOTE: It is assumed that the size of the subprograms is **smaller** than the size of the original problem.

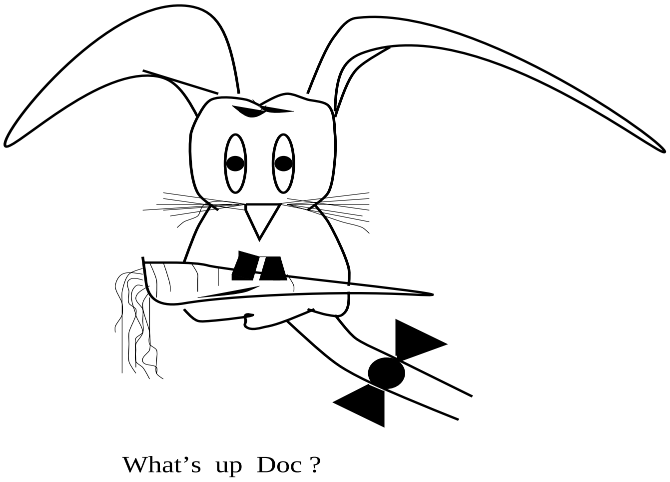

Figure 2:

1 pair of rabbits
after 1 month rabbits become fertile : 1 pair produces newborn pair.
(these rabbits never die & always reproduce each month)
month 0 : 1 pair
month 1 : 1 pair
month 2 : 1 + 1 pair
month 3 : 2 + 1 pair
month 4 : 3 + 2 pair
...
month n : month(n-1) + month(n-2) pairs

$F_n = F_{n-1} + F_{n-2}$
$F_0 = F_1 = 1$
1,1,2,3,5,8,13,21,34,55, ... ← **Fibonacci Numbers**

---

expontial growth : $F_n \geq 2 * F_{n-2}$
(Population more than doubles in two months)

FIBONACCI is an example of a
**LINEAR HOMOGENEOUS Recurrence Relation**

$$a_n = C_1 a_{n-1} + C_2 a_{n-2} + \ldots + C_r a_{n-r} \quad \text{(EQN 1)}$$

• A general solution of (1) involves a linear combination of individual solutions $a_n = \alpha^n$

•• Given r initial values this general solution can be made specific.

So substitute $a_k = \alpha^k$ in (1) to find the $\alpha$ 's

$$\alpha^n = C_1\alpha^{n-1} + C_2\alpha^{n-2} + \ldots + C_r\alpha^{n-r}$$

divide by $\alpha^{n-r}$

$$\alpha^r = C_1\alpha^{r-1} + C_2\alpha^{r-2} + \ldots + C_r \qquad \text{(EQN 2)}$$

(2) is called the **characteristic equation**
Assume no complex roots , we find r solutions $\alpha_1 \ldots \alpha_r$
Any linear combination of $\alpha' s$ is a *general solution*

$$\bullet a_n = A_1\alpha_1^n + A_2\alpha_2^n + \ldots + A_r\alpha_r^n$$

•• Initial values $a_1, \ldots, a_r$ determine a SPECIFIC Solution

---

Concepts related to *First Order*
Linear Homogeneous Recurrence Relations
Consider : $a_n = c_1 a_{n-1}$
This can be rewritten as
$\alpha^n = c_1\alpha^{n-1}$
If $S$ is a solution to $\alpha$, then $a_n = bS^n$
is a solution to the recurrence relation.

If $a_n = bS^n$
then we know
$a_0 = bS^0$ and so b = $a_0$.
And since $\alpha^n = c_1\alpha^{n-1}$ we know $\alpha = c_1$ is a solution
Therefore $a_n = b(c_1)^n$ and b = $a_0$

Example: Interest on Money at 20%
$A_n = A_{n-1} + .2(A_{n-1})$
$A_n = 1.2(A_{n-1})$
$A_n = 1.2 * 1.2(A_{n-2})$
$A_n = 1.2 * 1.2 * 1.2(A_{n-3})$
$A_n = (1.2)^x(A_{n-x})$
$A_n = (1.2)^n(A_0)$
where $A_0$ is the initial amount.

Another Example :

$$A_n = 3A_{n-1}; A_0 = 5$$
$$\alpha^n = 3\alpha^{n-1}$$
$$\alpha = 3$$
$$\text{So } A_n = b(3^n)$$
$$A_0 = b3^0 = b = 5$$
$$\text{Therefore } A_n = 5(3^n)$$
$$\text{In General } A_n = C * A_{n-1} = A_0 C^n$$

---

Concepts related to *Second Order*
Linear Homogeneous Recurrence Relations
Consider: $a_n = c_1 a_{n-1} + c_2 a_{n-2}$

- If $S_n$ and $T_n$ are solutions, then
$$a_n = bS_n + dT_n$$
is a solution.
- If r is a root of $\alpha^2 - c_1\alpha - c_2 = 0$ then
$r^n$ is a solution.

Roots MUST be solutions; if r is a root of

$$\alpha^2 - c_1\alpha - c_2 = 0$$

$$r^2 = c_1 r + c_2 \quad \Rightarrow \quad r^n = c_1 r^{n-1} + c_2 r^{n-2}$$

Use quadratic equation to find roots

$$\frac{-b \overset{+}{-} \sqrt{b^2 - 4ac}}{2a}$$

Given two unique roots
Then we obtain a GENERAL solution:

$$A(N) = b(r_1)^n + d(r_2)^n$$

To obtain a SPECIFIC SOLUTION, we need initial values to be able to solve for $b$ and $d$.

NEXT SHOW

$$a_n = bS^n + dT^n$$

Is a GENERAL solution to the recurrence relation
When $S^n$ and $T^n$ are solutions

Proof:
(1) $S_n$ and $T_n$ are solutions to $a_n = C_1 a_{n-1} + C_2 a_{n-2}$.
Hence, $S_n = c_1 S_{n-1} + c_2 S_{n-2}$ and $T_n = c_1 T_{n-1} + c_2 T_{n-2}$

*multiply the $S_n$ equation by $b$* *multiply the $T_n$ equation by $d$*

$$U_n = bS_n + dT_n = c_1(bS_{n-1} + dT_{n-1}) + c_2(bS_{n-2} + dT_{n-2})$$
$$U_n = c_1 U_{n-1} + c_2 U_{n-2}$$

Therefore, $U_n$ is a solution.

Given the initial conditions
$U_0 = b + d = K_0$
$U_1 = br_1 + dr_2 = K_1$
multiply the first equation by $r_1$ and substract
$br_1 + dr_1 = K_0 r_1$
-$br_1 - dr_2 = -K_1$
Hence, d$(r_1 - r_2) = r_1 K_0 - K_1$
Since $r_1 \neq r_2$
d = $\frac{r_1 K_0 - K_1}{r_1 - r_2}$
b = $\frac{K_1 - dr_2}{r_1}$
solve to show
$U_0 = b + d = K_0$
$U_1 = br_1 + dr_2 = K_1$

**Slide Lecture 1 –35**

**back to BUGS :**
$F_n = F_{n-1} + F_{n-2}$ $\qquad F_0 = F_1 = 1$
substitute $F_k = \alpha^k$
$\alpha^n = \alpha^{n-1} + \alpha^{n-2} \rightarrow \alpha^2 = \alpha + 1$
$\alpha_{1,2} = 1/2(1 \pm \sqrt{5})$ root of $\alpha^2 - \alpha$ - 1 = 0

$$F_n = b\left(\frac{1+\sqrt{5}}{2}\right)^n + d\left(\frac{1-\sqrt{5}}{2}\right)^n$$

*General Solution; initial values determine b & d*

**Slide Lecture 1 –36**

Solving Fibonacci for b and d

$$bS_0 + dT_0 = b + d = 1$$

$$bS_1 + dT_1 = b(1+\sqrt{5})/2 + d(1-\sqrt{5})/2 = 1$$

$$(COMMENT f_0 = 1 \Rightarrow b + d = 1 \text{ since } \ n^0 = 1)$$

$$f_1 = 1 \Rightarrow \left[ b(1+\sqrt{5})/2 + d(1-\sqrt{5})/2 = 1 \right]$$

$$f_1 = b(1+\sqrt{5}) + d(1-\sqrt{5}) = 2$$

---

Solve this pair of equations:

$$b(1+\sqrt{5}) + d(1-\sqrt{5}) = 2$$

$$-b(1-\sqrt{5}) - d(1-\sqrt{5}) = -(1-\sqrt{5})$$

$$b + b\sqrt{5} - b + b\sqrt{5} = 2 - (1-\sqrt{5})$$

$$2b\sqrt{5} = 1 + \sqrt{5}$$

$$\text{Therefore: } \ b = \frac{1}{\sqrt{5}}(\frac{1+\sqrt{5}}{2})$$

$$\text{Similarly: } \ d = -\frac{1}{\sqrt{5}}(\frac{1-\sqrt{5}}{2})$$

*Specific Solution*

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

When there is only ONE ROOT
Since r is the only solution to

$$Z^2 - c_1 Z - c_2 = 0$$

$$Z^2 - c_1 Z - c_2 = (Z - r)^2$$

solving :

$$c_1 = 2r, c_2 = -r^2$$

Also note

$$(r^2 - c_1 r - c_2) \Rightarrow (c_1 = c_2/r + r)$$

**Slide Lecture 1 –39**

Prove $nr^n$ is a solution (Induction/Substitution)
$2r^2 = c_1 r^1 + c_2 0.r^0$
$2r^2 = c_1 r^1 = 2r^2$
If $nr^n$ is a solution :

$$nr^n = c_1(n-1)r^{n-1} + c_2(n-2)r^{n-2}$$

$$nr^n = c_1 n r^{n-1} - c_1 r^{n-1} + c_2 n r^{n-2} - c_2 2r^{n-2}$$

$$\text{Since we know } \; c_1 = 2r, c_2 = -r^2$$

$$nr^n = 2nr^n - 2r^n - nr^n + 2r^{n2}$$

$$nr^n = 2nr^n - nr^n = nr^n$$

QED

**Slide Lecture 1 –40**

- If both roots are equal to r then,

let $a_0 = K_0$ and $a_1 = K_1$
In this case :

$$a_n = br^n + dnr^n$$
$$a_0 = br^0 + dnr^0 = b + dn = K_0$$
$$a_1 = br^1 + dnr^1 = br + dnr = K_1$$

Again we solve for b and d.

EXAMPLE

$$A_n = 4A_{n-1} - 4A_{n-2}; A_0 = A_1 = 1$$
$$\alpha^2 - 4\alpha + 4 = 0$$

r=2 Which Implies : $S_n = 2^n$ and $T_n = n2^n$

$$A_n = bS_n + dT_n = b2^n + dn2^n$$
$$A_0 = b = 1$$

$A_1 = b2 + d2 = 1$ Which Implies: $d = -\frac{1}{2}$

Therefore : $A_n = 2^n - n2^{n-1}$

LINEAR **IN** HOMOGENEOUS RECURRENCE RELATION
= Linear homogeneous + Term
e.g. $a_n = c.a_{n-1} + f(n)$ ..... (1)
Method :(1) Find Solution for the homogeneous part
in this case, $\beta c^n$ *(e.g.* $a_0 * c^n$*)*
(2) Suppose $\alpha_n^*$ is a *particular solution* of (1)
(i.e. $\alpha_n^*$ = c $\alpha_{n-1}^*$ + f(n))
then, $a_n = \beta c^n + \alpha_n^*$ is GENERAL SOLUTION
*general solution = (general homogeneous + PARTICULAR)*

Let's check this by substitution :
$a_n = \beta c^n + \alpha_n^* = c\beta c^{n-1} + c\alpha_{n-1}^* + f(n)$
= $c(\beta c^{n-1} + \alpha_{n-1}^*)$ + f(n)
= $ca_{n-1}$ + fn

OK

**Some Particular Solutions for** $a_n = Ca_{n-1}$ **+ f(n)**

When $C = 1, \Rightarrow a_n = \sum_{k=1}^{n} f(k) + a_0$

c $\neq$ 1 $\Rightarrow$ TABLE of PARTICULARS

| *f(n)* | *form of $\alpha$* |
|---|---|
| d (constant) | B (constant) |
| d.n | $B_1 n + B_2$ |
| $dn^2$ | $B_2 n^2 + B_1 n + B_0$ |

EXAMPLE : Towers of HANOI.
Complexity in # moves :
$a_n = 2a_{n-1} + 1$
homogeneous solution : $A2^n$
particular solution : B
B is a particular Solution so
$B = \alpha_n^* = 2\alpha_{n-1}^* + 1 =$ 2B + 1
B = -1
general solution $a_n = A2^n$ -1
initial value $a_1 = 1 \rightarrow A = 1$
Specific solution $\rightarrow a_n = 2^n - 1$
$a_1 = A2^1 - 1$
*1 = 2A - 1*
*2 = 2A*
*1 = A*

Another Example of a
Linear Inhomogeneous Recurrence Relation

$$a_n = 2a_{n-1} + n$$

$$a_0 = 0, a_1 = 1$$

$(a_1 = 1, a_2 = 4, a_3 = 11 \ ...)$

Homogeneous Part

$$a_n = 2a_{n-1}$$

$$a_n = 2^n$$

By Table look-up, Particular Form is $\beta N + \delta$
So, $a_{n-1} = \beta(n-1) + \delta$

Hence,

$$a_n = 2a_{n-1} + N$$

$$a_n - 2a_{n-1} = N$$

$$(\beta N + \delta) - 2(\beta(N-1) + \delta) = N$$

$$(\beta N + \delta) - 2\beta(N-1) - 2\delta = N$$

$$\beta N + \delta - 2\beta N + 2\beta - 2\delta = N$$

$$\beta N - 2\beta N + \delta - 2\delta + 2\beta = N$$

$$-\beta N - \delta + 2\beta = N$$

$$-\beta N^1 + (2\beta - \delta)N^0 = 1N^1 + 0N^0$$

Thus

$$-\beta N^1 = 1N$$

$$-\beta = 1 \quad \text{thus} \quad \beta = -1$$

And

$$(2\beta - \delta)N^0 = 0N^0$$

$$(2\beta - \delta) = 0$$

$$\delta = 2\beta$$

$$\delta = -2$$

So

$$\beta N + \delta = -N - 2$$

Therefore, General Solution $a_n = A.2^n + (-n - 2)$
solve for A
$a_0 = 0 = A - 2$
Therefore, A = 2
specific $a_n = 2.2^n - n - 2 = 2^{n+1} - n - 2$
check $a_0 = 0, a_1 = 1, a_2 = 4, a_3 = 11, \ldots$ OK