Chapter 4-

## Part 2: Advance counting Techniques :-

### Recurrence Relations :-

→ The no of bacteria in a colony doubles every hour

→ If a colony begins with five bacteria, how many will be present in $n$ hours?

→ Because the no of bacteria doubles every hour the relationship $a_n = 2a_{n-1}$ holds whenever $n$ is a positive integer

→ This relationship ($a_n = 2a_{n-1}$) holds together with condition $a_0 = 5$ uniquely determines $a_n$ for all non negative integer $n$.

→ We can find formula for $a_n$ from this information

→ Hence some counting problems can be solved by finding relationships, called recurrence relations between terms of a sequence.

### Recurrence relation :-

→ A recurrence relation for sequence $\{a_n\}$ is an equation that expresses $a_n$ in terms of one or more of previous terms of sequence namely $a_0, a_1, \ldots a_{n-1}$ for all integers $n$ with $n \geq n_0$ where $n_0$ is non-negative integer.

→ A sequence is called a solution of a recurrence relation if its term satisfy the recurrence relation

Examples:-

1) Let $\{a_n\}$ be sequence of recurrence relation and suppose that $a_n = a_{n-1} - a_{n-2}$ for $n = 2, 3, 4$ and suppose that $a_0 = 3$, $a_1 = 5$, what are $a_2, a_3$

Given recurrence relation

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

$a_2 = ?$ $\quad a_2 = a_1 - a_0 = 5 - 3 = 2.$

$a_3 = ?$ $\quad a_3 = a_2 - a_1 = 2 - 5 = -3$

similarly we can find $a_4, a_5$

∴ sequence we can find $a_4, a_5$

∴ sequence of recurrence relation solution of recurrence relation $= 3, 5, 2, -3 \ldots\ldots$

2) Determine whether sequence $\{a_n\}$ where $a_n = 3n$ for every non negative integer $n$ is a solution of recurrence relation $a_n = 2a_{n-1} - a_{n-2}$ for $n = 2, 3, 4 \ldots$

Given Recurrence relation

$$a_n = 2a_{n-1} - a_{n-2} \text{ for } n = 2, 3, 4 \ldots$$

$$a_n = 2[3(n-1)] - 3(n-2) \quad [\text{where } a_n = 3n]$$

$$a_n = 6n - 6 - 3n + 6$$

$$\boxed{a_n = 3n}$$

The sequence $\{a_n\}$ where $a_n = 3n$ is a solution of given recurrence relation

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

3) Determine whether sequence $\{a_n\}$ where $a_n = 2^n$ for every non negative integer, $n$ is a solution of recurrence relation $a_n = 2a_{n-1} - a_{n-2}$ for $n = 2, 3, 4$ ... Answer the same question where $a_n = 5$

$a_n = 2^n$

$a_0 = 1 \quad a_1 = 2, \quad a_2 = 4$

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

$a_2 = 2a_1 - a_0$

$a_2 = 3$

we got $a_2 = 4$ and with recurrence relation we got $a_2 = 3$

sequence $a_n = 2^n$ does not satisfy recurrence relation.

Given $\boxed{a_n = 5}$

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

for sequence $a_n = 5$, it is that
$a_1 = 5$
$a_2 = 5$
$a_3 = 5$
⋮
$a_n = 5$

$a_2 = 2a_1 - a_0 \Rightarrow \boxed{a_2 = 5}$

$a_n = 5$ the sequence (solution of recurrence relation)

satisfies given recurrence relation $a_n = 2a_{n-1} - a_{n-2}$

Recursive Algorithm :
It provides solution of a problem of size n in terms of solutions of one or more instance of same problem of smaller size.

Recurrence Relation : when we analyze the complexity of a recursive algorithm, we obtain recurrence relation that expresses the no of operations to solve a problem of size n

Modeling with Recurrence Relation :-

Modeling with Recurrence Relation

1) compound interest - Recurrence Relation
Suppose that a person deposits \$10,000 in a savings account at a bank yielding 11% per year with interest compounded anually. How much will be in the account after 30 years.
To solve this problem
Let $P_n$ denote the amount in account after n years
→ The amount in account after n years equals the amount in account after n-1 years plus interest for $n^{th}$ year. We see that recurrence relation

$$P_n = P_{n-1} + 0.11\,P(n-1)$$

$$P_1 = (1.11)\,P_0$$

$$P_2 = (1.11)\,P_1 = (1.11)^2\,P_0$$

$$P_3 = (1.11)^3\,P_0$$

$$P_n = (1.11)^n\,P_0$$

after 30 years $P_{30} = (1.11)^{30} \times 10{,}000 = \$228{,}922.87$

Scanned with CamScanner

Rabbits and fibonacci numbers-

1) Recurrence Relation.

consider this problem which was orginally posed by leonardo pesano, also known as fibonacci in 13th century in this book liberabori

"A young pair of rabbits (one of each sex) is placed on an island. A pair of rabbit does not breed until they are 2 months old. After they another pair each month as shown find recurrence relation for no of pairs of rabbits on island after n months, assuming that no rabbits ever die.

sol: Let $f_n$ = No of pairs of rabbit after n months

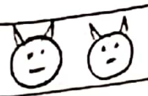
$f_n$, n = 1, 2, 3

→ Rabbit population can be modeled using recurrence relation

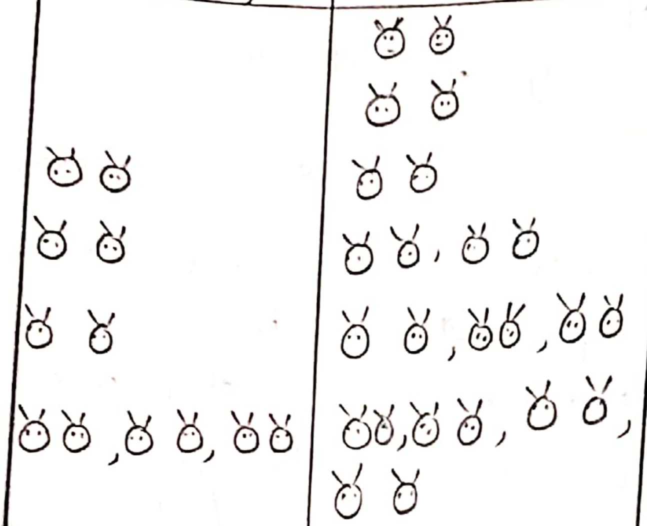
At end of 1st month, the no of pair of rabbit on island is $f_1 = 1$

This is because this place doesnot breed during the second month $f_2 = 1$

To find no of pair after n months add the no of island the previous month of $f_{n-1}$ and no of new born pair which equal $f_{n-1}$ because each newborn pair comes from atleast 3 months old.



denotes pair of rabbits



| Reproducing pair (atleast 2 months old) | Young pair (Less than 2 months old) | Month | Reproducing Pair | Young Pair | Total Pairs |
|---|---|---|---|---|---|
| | | 1 | 0 | 1 | 1 |
| | | 2 | 0 | 1 | 1 |
| | | 3 | 1 | 1 | 2 |
| | | 4 | 1 | 2 | 3 |
| | | 5 | 2 | 3 | 5 |
| | | 6 | 3 | 5 | 8 |

consequently the sequence $\{f_n\}$ satisfies the recurrence relation

$$f_n = f_{n-1} + f_{n-2}$$

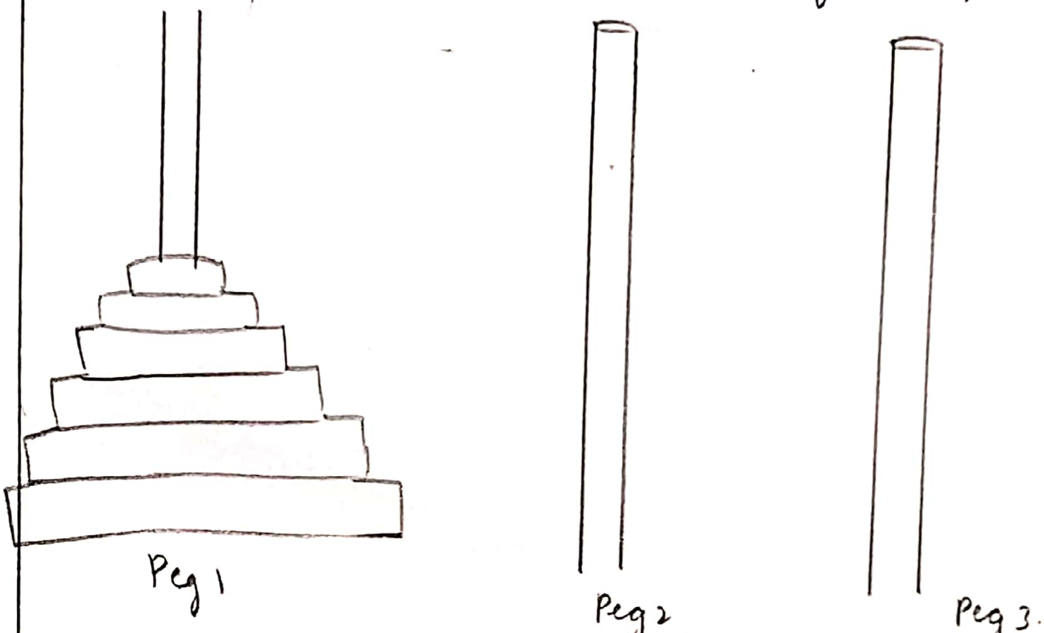
for $n \geq 3$ together with intial conditions $f_1 = 1$ & $f_2 = 1$. This recurrence relation uniquely determine sequence, which is fibonacci sequence

The tower of hanoi:

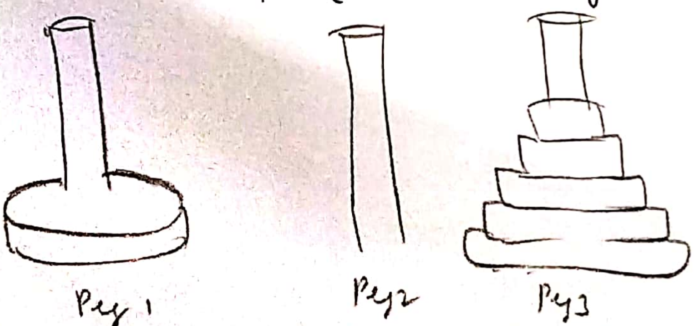
A popular puzzle of late Nineteenth century invented by french mathematician edward lucas called towers of hanoi consists of three pegs mounted on a board together with disks are placed on first peg in order of six with largest on bottom. The rules of puzzle allow disks to be moved one at a time from one peg to another as long as disk is never

placed on top of smaller disk.
The goal of puzzle is to have all disks one second peg order of size, with largest on bottom.

Let $H_n$ be no of moves needed to solve the tower of hanoi problem with $n$ disks. Setup recurence relation for sequences $\{H_n\}$.




* Begin with $n$ disks one peg 1.
* We can transfer the top $n-1$ disks, following rulers of puzzle to peg 3 using $H_{n-1}$ moves




* Now use one move to transfer the largest disk to the second peg
* We can transfer disks on peg 3 to Peg 2 using $H_{n-1}$ addition al moves, placing them on top of largest disk.

This moves sets up recurrence relation

$H_n = H_{n-1} + 1 + H_{n-1}$

$H_n = 2H_{n-1} + 1$

[ $H_1 = 1$ because one disk can be transfers from peg 1 to peg 2 according to rules of puzzle)

We can use iterative approach to solve recurrence relation

$H_n = 2H_{n-1} + 1$

$H_n = 2^2(2H_{n-2} + 1) + 1$

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

$H_n = 2^3 H_{n-3} + 2^2 + 2 + 1$

$= 2^{n-1} H_{n-(n-1)} + 2^{n-2} + \ldots 2^2 + 2 + 1$

$H_n = 2^{n-1} H_1 + 2^{n-2} + \ldots + 2^2 + 2 + 1$

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

$\boxed{H_n = 2^n - 1}$

Find a recurrence relation and give initial conditions for the numbers of bit strings of length n that does not have two consective 0's. How many such bit strings are there of length five

Sol:

$a_n$ - number of bit strings of length n that do not have consective 0's

$a_0 = 0$

No of strings come under $a_0$

0

$a_0 = 0$

Note: There exists no strings of length 0

string come under $a_1$

0

1

$a_1 = 2$

$a_2 = 3$

strigs come under $a_2$

01

10

11

$a_3 = 5$

strings came under $a_3$

010

011

101

110

111

$a_4 = 8$

strings came under $a_4$

0101

0110

0111

1010

1011

1100

1110

1111

$a_3 = a_2 + a_1 = 3+2 = 5$
$a_4 = a_3 + a_2 = 5+3 = 8$
with above relations we can conclude

$a_n = a_{n-1} + a_{n-2}$ for $n \geq 3$

Codeward Enumeration :-
A computer system consider a string of decimal digits a valid codeward if is valid
For instance · 1230407869 is not valid. det whenever 120987045608
$a_n$ be the number of valid n-digit codewards

Find a recurrence relation for $a_n$

$a_n$ = No of valid n-digit codewards.
$a_1 = 9$ [From 10 digits (0 to 9) are valid 1-digit codewards]

$a_n$ = Previous valid $(n-1)$ digit codewards by digits (1 to 9) + invalid $(n-1)$ digit codewards appended by 0'

So we take

$\boxed{a_{n-1} = \text{No of valid } (n-1) - \text{digit codeward}}$

case 1:- To make valid $(n-1)$ digit codeward to valid n-digit code word it is append "digit' from 1 to 9 digits Two can be done by 9

$\therefore$ $9a_{n-1}$ gives us valid n-digit code
$\rightarrow (1)$

case 2. we take invalid $(n-1)$ digit strings we extract invalid $(n-1)$ digits from total string of length $(n-1)$ by subtraction valid $(n-1)$ digits

ie $10^{n-1}$ = No of Total Strings of length $(n-1)$

No of invalid $(n-1)$ digit string = No of Total string of length $(n-1)$ − Valid $(n-1)$ length String

$$= (10^{n-1} - a_{n-1}) \quad -②$$

This invalid Strings of length $(n-1)$ are made valid $n$-length strings by appending 'zero' This can be done only one way

Add ① & ② from case 1 & case 2 to get $a_n$

$$a_n = 9 \times a_{n-1} + (10^{n-1} - a_{n-1})$$

$$\boxed{a_n = 8a_{n-1} + 10^{n-1}}$$

Scanned with CamScanner

Find a recurrence relation for $C_n$, the no of ways to paranthesize the product of $n+1$ numbers $x_0, x_1, x_2, \ldots x_n$ to specify the order of multiplication

for example $C_3 = 5$ because there are five ways to paranthesize $x_0, x_1, x_2, x_3$ to determine the order of multiplication

$$((x_0 \cdot x_1) \cdot x_2) \cdot x_3$$

$$(x_0 \cdot (x_1 \cdot x_2)) \cdot x_3$$

$$(x_0 \cdot x_1) \cdot (x_2 \cdot x_3)$$

$$x_0((x_1 \cdot x_2) \cdot x_3)$$

$$x_0 \cdot (x_1 \cdot (x_2 \cdot x_3))$$

The Recurrence relation for $(n+1)$ numbers is $C_n = C_0 C_{n-1} + C_1 C_{n-2} + \cdots C_{n-2} C_1 + C_{n-1} C_0$.

$$\boxed{C_n = \sum_{k=0}^{n-1} C_k C_{n-k-1}}$$

intial condition

$C_0 = 1$

$C_1 = 1$

The sequence $\{C_n\}$ is the sequence of catalon numbers

$C_1 = C_0 C_0 = 1$

$C_2 = C_0 C_1 + C_1 C_0 = 2$

$C_3 = C_0 C_2 + C_1 C_1 + C_2 C_0$

$= 2 + 1 + 2$

$= 5$