Page No.
Date : / /

* Stack:

Stack is a linear Data structure, in which we perform the Insertion and Deletion Operation.

Stack is follow LIFO structure. It means Last In First Out.

In stack, whose data enter first this is remove by last.

-> Representation of Stack in Memory:

A stack representation using array method.

-> Stack Operation:

Stack is follow This Four Operation.

1) Push
2) Pop
3) Peep
4) Change or Update.

1 Push Operation: When an element insert in a stack, then Push Operation is use.

Brain Shot

SMVS
Page No.
Date : / /

To Perform the Push Operation we have to check "Stack Overflow" Condition.

- Stack Overflow: Before the insert the element, we have to check stack is full or empty.

If Stack is Full, then we cannot insert the element in stack.

If stack is empty, then we can insert the element in stack.

- Algorithm of Push Operation:

1 Start
2 If top >= n
{ print "Stack Overflow" }
return

3 Set top = top + 1
4 S[top] = data
5 Return.

Where top = Index of Stack

Brain Spot

2 Pop Operation : When an element delete in a stack, then pop operation is use.

To Perform the Pop Operation, we have to check "stack Underflow" condition.

- Stack Underflow : Before deleting the element in stack, we have to check stack is empty or not

If stack is empty, then we can not delete element in a stack.

If Stack is not empty, then we can delete element in a stack.

- Algorithm of Pop Operation.

1 Start
2 If top = -1
    print "Stack Underflow"
    Return.
3 Data = S [top]
4 Set top = top - 1
5 Exit

Brain Spot

3 Peep Operation: When an element display in a stack, then peep operation is use.

To perform the Peep Operation, we have to check "stack Underflow" condition.

- Stack Underflow: Before display the element in stack, we have to check stack empty or not.

If Stack is empty, then we cannot display element in a stack.

If Stack is not empty, then we can display element in a stack.

- Algorithm of Peep Operation:

1 If top = -1
print "Stack is Empty"
Return

2 else, set item = stack(Top-i+1)

3 S[top] = Data

4 End.

Brain Spot

4 Change Operation: When an element change or update in a stack, then change Operation is use.

To Perform the change Operation. We have to check "stack Underflow" condition.

- Stack Underflow: Before change the element in stack, we have to check stack empty or not.

If stack is empty, then we can not update the element.

If stack is not empty, then we can update the element.

- Algorithm:

1 Start
2 If top = -1
    print "Stack is empty"
    Return
3 set stack [top-i+1] = data
4 End

Brain Spot

- Application of Stack:

1 Expression Evaluation

2 Reversal of a String

3 Conversion of infix expression to postfix forms.

4 Tower of Hanoi.

- Expression Evaluation:

1 InFix Expression: Operators are placed in between the Operands.

2 PreFix Expression: Operators are placed before the operands.

3 Postfix Expression: Operators are placed after the operands.

- Conversion of infix expression to Postfix Forms

Ex (A + B / C * (D + E) - F)

Pram Spot

... શાંતિ અને સંતોષ અનુભવાય છે.

InFix	Stack	PostFix
C	(	
A	(	A
+	(+	A
B	(+	AB
/	(+/	AB
C	(+/	ABC
*	(+*	ABC/
C	(+*C	ABC/
D	(+*C	ABC/D
+	(+*C+	ABC/D
E	(+*C+	ABC/DE
)	(+*C*)	ABC/DE+
-	(*-	ABC/DE+*+
F	(-	ABC/DE+*+F
)	(-)	ABC/DE+*+F-

- Algorithm:

1 Add the symbol '(' into stack ')' at the end of array index.

2 Write the symbol of array infix one by one from left to right.

3 IF symbol is left side '(' then add it to the stack.

Brain Spot

વાંચવા-વિચારવા કરતાં વર્તનમાં ઉતારવું વધુ સારું

4 If symbol is operands, then add into postfix.

5 If Symbol is operator,

(1) Then Poped the operator which have same or higher Priority.
(2) Add the poped operator in Postfix.
(3) Add the scanned operator in Stack.

6 If symbol is ')' than pop all the operators in stack until symbol is '(' in stack and Remove symbol From stack.

7 Follow this step untill the infix into ends.

8 End.

- Tower of Hanoi:

There are three rules for Tower of Hanoi.

1 Only one disk move at a time.

Brain Spot

Page No.
Date : / /

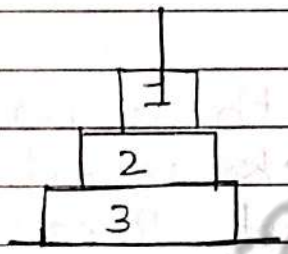
2 First only top disk can be moved.

3 A larger disk can not be moved on a smaller disk.

- If Disk = n,

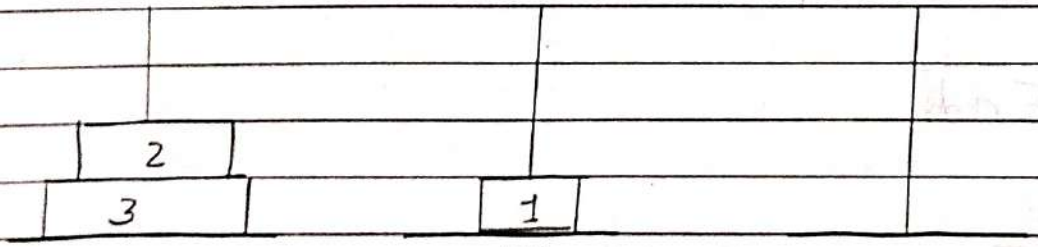
then number of moves = $2^n - 1$

Ex.

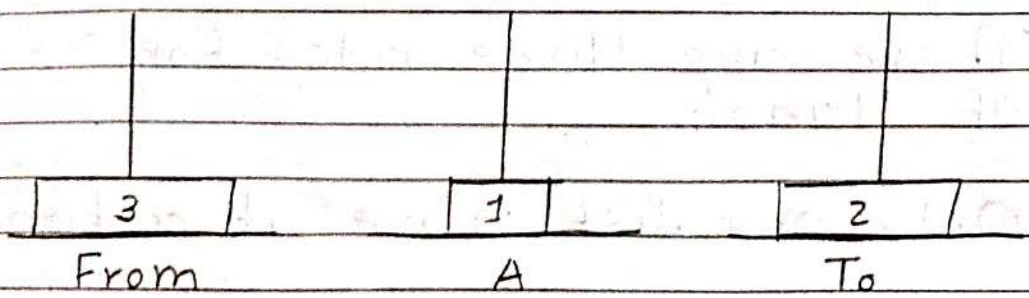


Disk = 3, then moves = $2^3 - 1$
= 7

1

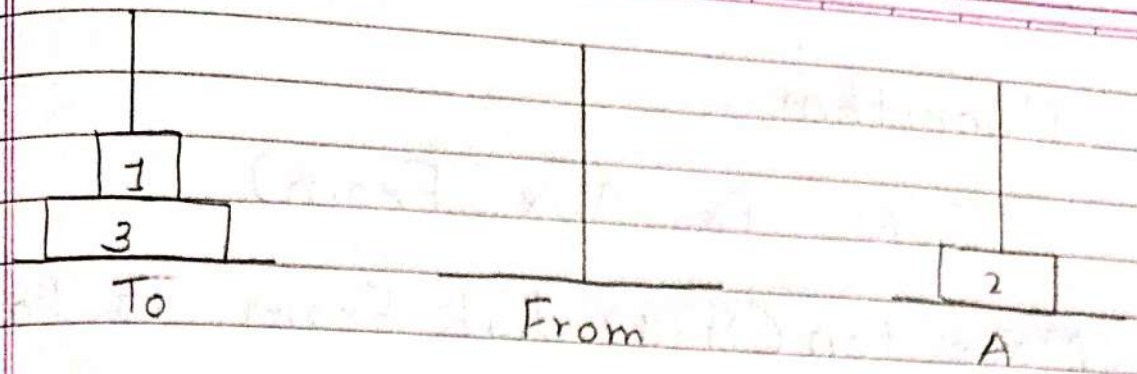


2

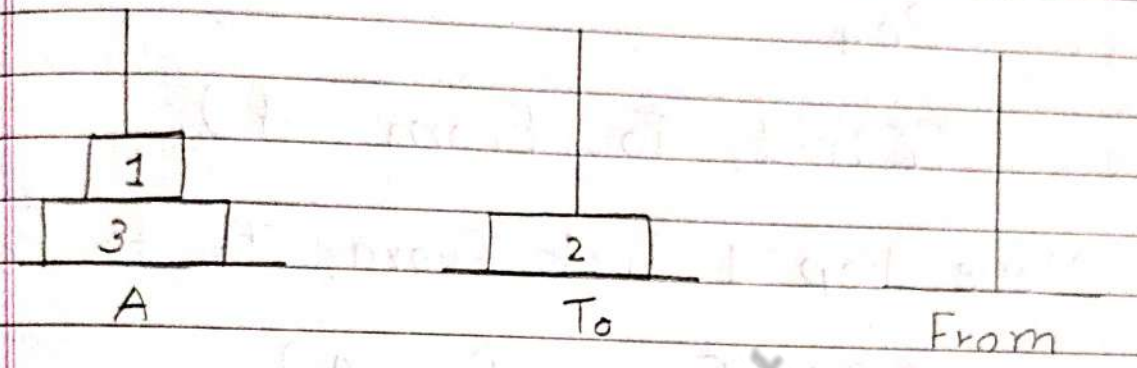


Brain Spot

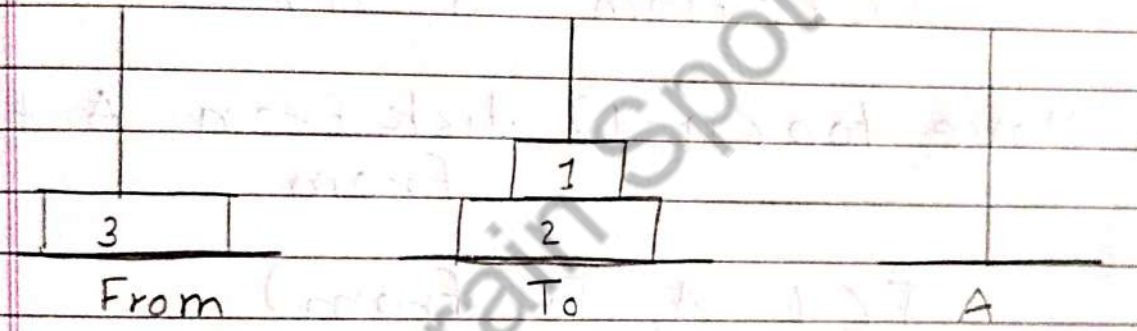
3



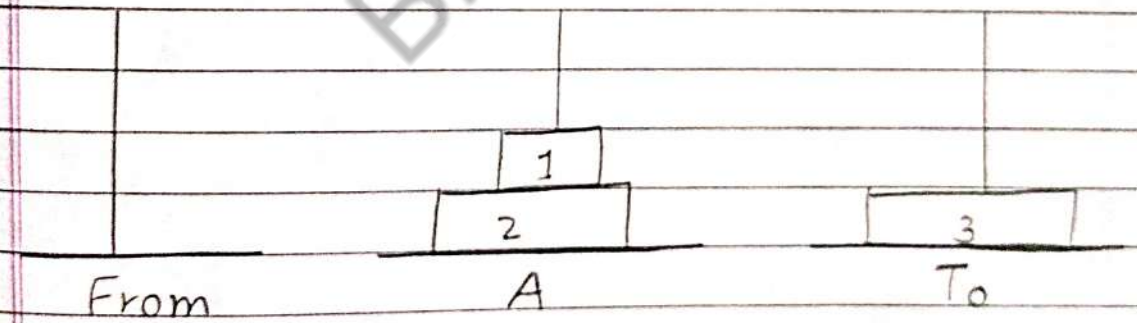
4



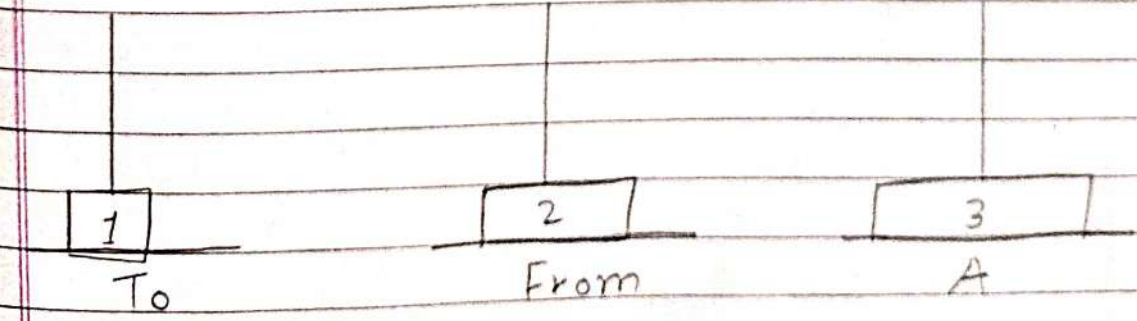
5



6



7



Brain Spot

SMVS
Page No.
Date : / /

- Algorithm:

T(N, To, Aux, From)

1 Move top (n-1) disk From, To to A Peg

T(n-1, To, From, A)

2 Move top 1 disk from, To to From

T(1, From, To, A)

3 Move top (n-1) disk From, A to from.

T(1, A, To, from)

Brain Spot