**Tower of Hanoi**

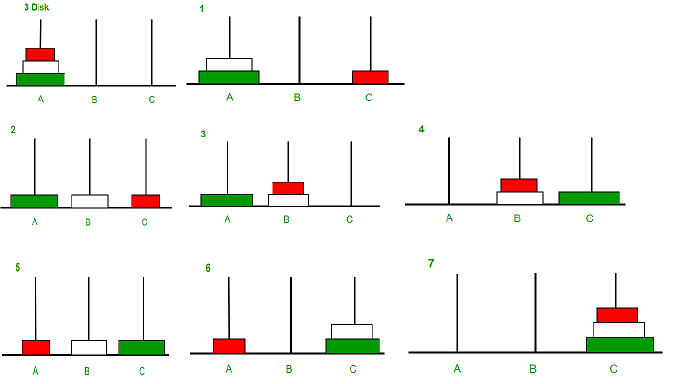
Tower of Hanoi is a mathematical puzzle where we have three rods and n disks. The objective of the puzzle is to move the entire stack to another rod, obeying the following simple rules:

1) Only one disk can be moved at a time.

2) Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack

i.e. a disk can only be moved if it is the uppermost disk on a stack.

3) No disk may be placed on top of a smaller disk.

In figure 1:

n=3;

move disk from rod A->C

then move from rod A->B

then move from rod C->B

then move from rod A->C

then move from rod B->A

then move from rod B->C

then move from rod A->C

use function to implement is TOWER(n,beg,aux,end)

where n i sthe no. Disks, beg, end ar rods and aux is helping rod like rod A,C and aux is rod B. Motive is to shift all n from beg->end.where n>1.

TOWER(n-1,beg,end,aux)

TOWER(1,beg,aux,end)

TOWER(n-1,aux,beg,end)

above are the three functions of recursively solving the tower of hanoi.

e.g

n=4

TOWER(4,A,B,C) WHERE B IS AUXILARY

DISK MOVES WILL BE 15

1.e f(n)=2^n-1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | TOWER(1,A,C,B) | A->B |
|  |  | TOWER(2,A,B,C) | A->C |  |
|  |  |  | TOWER(1,B,A,C) | B->C |
|  | TOWER(3,A,C,B) | A->B |  |  |
|  |  |  | TOWER(1,C,B,A) | C->A |
|  |  | TOWER(2,C,A,B) | C->B |  |
|  |  |  | TOWER(1,A,C,B) | A->B |
| TOWER(4,A,B,C) | A ->C |  |  |  |
|  |  |  | TOWER(1,B,A,C) | B->C |
|  |  | TOWER(2,B,C,A) | B->C |  |
|  |  |  | TOWER(1,C,B,A) | C->A |
|  | TOWER(3,B,A,C) | B->C |  |  |
|  |  |  | TOWER(1,A,C,B) | A->B |
|  |  | TOWER(2,A,B,C) | A->C |  |
|  |  |  | TOWER(1,B,A,C) | B->C |

**#include<iostream>**

**using namespace std;**

**void TOWER(int n,char BEG,char AUX,char END) {**

**if(n>0) {**

**TOWER(n-1,BEG,END,AUX);**

**cout<<BEG<<"->"<<END<<endl;**

**TOWER(n-1,AUX,BEG,END);**

**}}**

**int main() {**

**int n=4;**

**TOWER(n,'A','B','C');**

**return 0;**

**}**

**//with class without constructor**

**#include<iostream>**

**using namespace std;**

**class temp{**

**public:**

**void TOWER(int n,char BEG,char AUX,char END) {**

**if(n>0) {**

**TOWER(n-1,BEG,END,AUX);**

**cout<<BEG<<"->"<<END<<endl;**

**TOWER(n-1,AUX,BEG,END);**

**}}**

**};**

**int main() {**

**temp p;**

**int n=4;**

**p.TOWER(n,'A','B','C');**

**return 0;**

**}**

**//with constructor**

**#include<iostream>**

**using namespace std;**

**class TOWER{**

**public:**

**TOWER(int n,char BEG,char AUX,char END) {**

**if(n>0) {**

**TOWER(n-1,BEG,END,AUX);**

**cout<<BEG<<"->"<<END<<endl;**

**TOWER(n-1,AUX,BEG,END);**

**}}**

**};**

**int main() {**

**TOWER p(n,'A','B','C');**

**return 0;**

**}**