Torres de Hanoi:

Ao executar o código das torres de hanoi o sistema solicita a quantidade de discos desejados para a torre, ao informar o número de discos o sistema mostra o número minimo de movimentos que devem serem realizados para mover todos os discos do pino A para o pino C sem que um disco maior fique sobre o disco menor, isso é possivel pela técnica de recursividade onde o metodo **void moverDiscos(int n, char a,char c,char b)** é chamado varias vezes dentro do seu proprio corpo de estrutura, exemplo:

void moverDiscos(int n, char a,char c,char b){

if(n==1){

System.out.printf("Mover um disco de %c para %s\n",a,c);

nmov++;

}else{

//chamada do metodo abaixo:

**moverDiscos(n-1, a,b,c);**

**moverDiscos(1,a,c,b);**

**moverDiscos(n-1,b,c,a);**

}

}

O procedimento de recursividade é usado para realizar os movimentos dos pinos sem que um pino maior fique sobre um pino menor.

?

Int s[];

Int nr;

Int nsol;

Int col;

Queens(int n){

nr=n ;

s=new int[nr+1];

nsol=0;

col=0;

}

Void escreverSolucao(){

System.out.printf(“Solução(%2d): (”,nsol);

For(int i=1; i<=nr-1; i++){

System.out.printf(“%d,”,s[i]);

System.out.printf(“%d)\n”,s[nr]);

}

Boolean atacada(int l,int c){

Int I;

Boolean emPerigo=false;

I=1;

While(i<c && ! emPerigo){

emPerigo=(s[i]==l) ll (Math.abs(s[i]-l)== Math.abs(i-c));

i++;

}

Return emPerigo;

}