* A Torre de Hanoi é um jogo que tem três torres, e uma delas tem discos em ordem de tamanho, onde a base é o maior e o topo o menor.

Para vencer é preciso transportar todos os discos de uma torre para outra.

* As regras são:
  + Apenas o disco que estiver no topo pode ser transportado.
  + Apenas um disco pode ser transportado por vez.
  + Nenhum disco pode estar em cima de outro que seja menor.

Para cada “n” discos, existe um valor mínimo de movimentações possíveis para que se possa vencer. Essa quantidade pode ser descoberta com o seguinte algoritmo em PORTUGOL:

Algoritmo torreHanoi

DECLARE

funcao minimoMov(num: inteiro): inteiro

INICIO

se (n <= 1) entao

retorne 1;

senao

retorne minimoMov(num - 1) \* 2 + 1;

fimse;

fimfuncao;

numero: inteiro;

INICIO

escreva("Numero de discos: ");

leia(numero);

escreva("Resultado usando a funcao recursiva: ", minimoMov(numero));

fimalgoritmo;

Ou simplesmente de forma direta: (2^n) - 1

* Também é possível descobrir a correta movimentação dos “n” discos, algoritmos que mostram isso:

Pseudocódigo usando o método recursivo:

Algoritmo TorreDeHanoiRecursivo

DECLARE

procedimento hanoi(n: inteiro, from, temp, to: caractere)

INICIO

se (n != 0) entao

hanoi(n-1, from, to, temp);

Escreva("Mova o disco ", n ," de ", from ," para ", to);

hanoi(n-1, temp, from, to);

fimse;

fimprocedimento;

INICIO

hanoi(4, 'A', 'B', 'C');

fimalgoritmo;

Linguagem de programação C++ usando método iterativo:

#include <iostream>

using namespace std;

int main() {

int n = 4, from, to, k;

for(k=1; k < (1 << n); k++) {

from = ( k & ( k - 1 ) ) % 3;

to = ( ( k | ( k - 1 ) ) + 1 ) % 3;

cout << "Mova o disco de " << from << " para " << to << endl;

}

}

* A diferença entre esses dois algoritmos são:
* No recursivo, existe um procedimento, que chama ele mesmo se “n” for diferente de 0. A cada chamada, o parâmetro “n”(um inteiro) é subtraído 1, fazendo que em algum momento o “n” chegue em 0(sua condição de parada), evitando também que o código entre em loop infinito. Além da subtração do “n”, a ordem do “from”, “to” e “temp” também é reposicionado, isso permite que mostre exatamente o disco de qual torre deve ser movido para qual torre.
* Já no método iterativo, é tudo executado dentro de um loop, usando a técnica do bitmap, onde “k” serve para calcular “from” e “to” a nível binário, de maneira que eles formam a ordem exata das movimentações, essas instruções são dadas a cada iteração, e o loop só para quando “k” for maior que 2^n – 1(quantidade mínima de movimento).

- O método recursivo é a melhor maneira pra os humanos compreender o que está acontecendo, simplesmente olhando para o algoritmo, porém, quanto maior for o numero de discos, maior será o custo computacional e também levará muito mais tempo para concluir a execução.

- E o método iterativo ocorre o contrário, é muito dificultoso para um humano entender o que está acontecendo no algoritmo, tendo que realizar varias análises para chegar à uma resposta, entretanto, o custo computacional é incrivelmente pequeno, mesmo para um grande valor de “n”, e o tempo de execução, mesmo que ele continua crescendo de acordo com o valor de “n”, ele cresce muito menos que o método recursivo.

Ou seja, se é necessário o entendimento dos processos, a melhor opção é usar o método recursivo, mas se o foco é obter desempenho e o menor custo possível, então a opção é usar o método iterativo.

Ivan Luiz Pivetta de Oliveira 802.495