Cópia de Relatório sobre o Paradigma Indução/Recursão em um Solver de Torre de hanoi em Python

Nome: Daniel Antônio de Sá **Disciplina: PAA**

O objetivo deste trabalho é apresentar um solucionador eficiente para o problema da Torre de Hanói e explicar como as técnicas de indução e recursão desempenham um papel fundamental na resolução deste quebra-cabeça.

Indução no Problema da Torre de Hanoi:

A indução é usada para provar que nosso algoritmo funciona corretamente para qualquer número de discos na Torre de Hanoi. No caso da Torre de Hanoi, podemos dividi-lo em duas partes:

Caso Base da Indução:

Começamos resolvendo o problema para o caso mais simples, que é quando temos apenas um disco. Nesse caso, mover o disco da haste de origem para a haste de destino é trivial e serve como nosso caso base. Provamos que nosso algoritmo funciona para um único disco.

Passo da Indução:

Suponha que nosso algoritmo funcione corretamente para n discos.

Usando essa suposição (hipótese de indução), mostramos que ele também funciona para n+1 discos.

Isso nos permite concluir que nosso algoritmo é válido para qualquer número de discos na Torre de Hanoi, usando o princípio da indução.

Recursão no Problema da Torre de Hanoi:

A recursão é uma técnica crucial na implementação do solucionador da Torre de Hanoi, pois o problema é naturalmente recursivo. A abordagem recursiva é a seguinte:

Para mover n discos da haste de origem para a haste de destino, usamos uma haste intermediária como auxiliar. Dividimos o problema em três etapas:

Mover os n-1 discos superiores da haste de origem para a haste auxiliar, usando a haste destino como auxiliar.

Mover o disco maior (n-ésimo) da haste de origem para a haste destino diretamente.

Mover os n-1 discos da haste auxiliar para a haste destino, usando a haste origem como auxiliar.

Essa abordagem divide o problema em subproblemas menores e resolve-os de maneira recursiva até que o caso base (um disco) seja atingido, momento em que o problema é trivial de resolver.

Para a solução do algoritmo, foi desenvolvido uma função em python que recebe como parâmetro o número de discos, torre de origem, torre de destino, e torre auxiliar. Como caso base, verificamos se o número de discos é maior que 0 para ir chamando a recursão, até que se iguale a zero e termine.

A lógica da chamada recursiva se dá pela seguinte forma:

a. Primeira chamada recursiva (self.solver(num - 1, torreO, torreT, torreD)):

Move num - 1 discos da Torre O (origem) para a Torre T (auxiliar), usando a Torre D (destino) como torre auxiliar. Isso é feito chamando o solver novamente com um disco a menos e trocando as torres de destino e auxiliar.

b. Mover o Disco (self.mover(torreO, torreD)): (dentro dessa função mostamos o movimento e o estado de cada torre).

Após mover num - 1 discos da Torre O para a Torre T, o algoritmo move o disco restante da Torre O para a Torre D. Isso é feito diretamente, pois é o único disco restante e pode ser movido sem problemas.

c. Segunda chamada recursiva (self.solver(num - 1, torreT, torreD, torreO)):

Finalmente, move os num - 1 discos que estão na Torre T (auxiliar) para a Torre D (destino), usando a Torre O (origem) como torre auxiliar. Isso é novamente feito chamando o solver recursivamente com as torres apropriadas.

Essa lógica se repete até que num chegue a 0, que é o caso base da recursão. Em cada nível de recursão, o algoritmo resolve um subproblema movendo num - 1 discos da origem para a auxiliar, movendo um disco da origem para o destino e, em seguida, movendo os num - 1 discos da auxiliar para o destino.

O uso da indução e da recursão no solucionador da Torre de Hanoi permite que o algoritmo resolva eficientemente o problema para qualquer número de discos. A indução é usada para provar a validade do algoritmo para todos os casos, enquanto a recursão é a estratégia principal de resolução do problema em si, dividindo-o em subproblemas gerenciáveis e permitindo a solução de forma iterativa. Essas técnicas combinadas tornam o nosso solucionador eficiente e escalável, sendo útil em uma variedade de aplicações que envolvem a manipulação de sequências ordenadas de elementos.