Capítulo 02 – Recursão

Dentro de Ciência da Computação, quando há uma rotina(função ou método) que se auto-invoca, dizemos então que temos uma rotina recursiva. Mas então, como tratar recursividade? Recursividade é um caminho/ferramenta abstrata que você via ultlizar (ou não) dependendo da situação para tratar problemas em que cada instância contém uma instância menor do mesmo problema(lógica recursiva).
Note que há uma desvantagem em usar esse tipo de algoritmo, pois ele utiliza bastante memória(se comparado a outros algoritmos), usando um grande número de pilhas.

2.1 Algoritmos Recursivos

Algoritmos recursivos resolvem problemas recursivos que possuem a lógica já citada, o algortimos segue a seguinte lógica:

***se** a instância em questão é pequena*
- resolva diretamente (use força bruta se nescessário) ;
senão,
- redusa-a a uma instância menor do mesmo problema;
- aplique aoo método á instância menor;
- e volte á instância original.

Nós chamamos de trivial a menor instância do problema.

2.2 Fibonacci

Um exemplo bem comum é o de encontrar um fatorial. Um fatorial pode ser escrito da seguinte maneira: n * n-1 * n-2 * n-3 * … * 2 * 1. Nosso trabalho aqui é encontrar uma instância menor, aplicar o método à essa instância menor e voltar a instância original.

Note o que acontece para n=4 :

4! = 4 * 3!
3! = 3 * 2!
2! = 2 * 1!
1! = 1 * 0!
0! = 1

Até esse ponto o computador foi buscando dentro de instâncias menores o valor. Apartir daqui como ele encontrou o valor da menor instância ele pode voltar “respondendo” a cada chamada, assim ele volta a instância original com a resposta:

4! = 4 * 3!
3! = 3 * 2!
2! = 2 * 1!
1! = 1 * 0!
0! = 1
1! = 1
2! = 2
3! = 6
4! = 24

Basicamente essa é a lógica recursiva. Vejamos mais exemplos:

2.3 Torre de Hanoi

2.4 Máximo Divisor Comum