Algoritmos recursivos

Introdução

Definição

- Uma função recursiva é aquela que se chama a si própria

Ideia fundamental

- Uma solução para um problema é definida em termos de uma solução análoga, mas mais simples do que ela própria

Funcionam da seguinte forma

- Uma função recursiva é chamada da mesma forma que uma função iterativa
- Quando é feito a chamada recursiva, as suas variáveis locais são colocadas na pilha ("stack") do sistema
- Quando a chamada recursiva retorna, a função que a chamou continua normalmente

Partes de uma função recursiva

1ª) Parte básica (caso terminal/base, ponto de paragem)

- Constitui a versão mais simples do problema para a qual a solução é conhecida
- Não usa a recursividade
- Corresponde ao limite superior ou inferior do caso geral

2ª) Parte recursiva (caso/passo geral, regra geral)

- O problema é definido em termos de uma versão mais simples de si própria (casos mais pequenos)

Desenho de uma função recursiva

Passos a realizar

- Obter uma solução de como o problema pode ser dividido em passos mais pequenos
- Definir o caso terminal (ponto de paragem)
- Definir o caso geral (regra geral)
- Verificar se o algoritmo termina

Tipos de recursividade

Direta

- A função contém uma chamada explícita a si própria
- O resultado final está associado à primeira chamada recursiva da função

Indireta

- A função F chama a função G que por sua vez chama F

Terminal

- A função contém uma chamada explícita a si própria
- O valor de retorno da função é o valor de retorno da última chamada recursiva (não existem operações pendentes)

Exemplo

Enunciado

factorial(n)

Forma (expressão) iterativa

factorial(n) = 1 x 2 x ... x (n-1) x n

Função iterativa

```
int  factorial (int  n)
{
  int  k, resultado;
  resultado = 1
  for (k = 1; k <= n; k++)
      resultado = resultado * k;
  return resultado;
}
```

Forma (expressão) recursiva

$$\text{factorial}(n) = \begin{cases} 1, & \text{se } n = 0 \\ n \times \text{factorial}(n-1), & \text{se } n > 0 \end{cases}$$

Função recursiva direta

```
int  factorial (int n)
{
  if (n == 0)
      // caso terminal / ponto paragem
      return 1;
  else
      // caso geral / regra geral
      return  n * factorial(n-1);
}
```

Função recursiva terminal

```
int  factorial (int n, int r)
{
  if (n == 0)
      // caso terminal / ponto paragem
      return r;
  else
      // caso geral / regra geral
      return  factorial(n – 1, n * r);
}
```