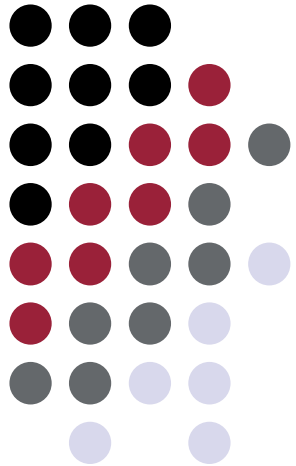
A12 Recursão

CSI030 – Programação de Computadores I

Prof. Dr. George H. G. Fonseca
Universidade Federal de Ouro Preto



Motivação

- Muitos problemas têm a seguinte propriedade: cada instância do problema contém uma instância menor do problema
- Dizemos que esses problemas tem estrutura recursiva
- Para resolver tal problema podemos aplicar o seguinte método:
  - Se o instância for pequena, resolva-a diretamente
  - Senão, reduza-o a uma instância menor do mesmo problema

Exemplo
Cálculo de Fatorial

- Considere a seguinte solução para o cálculo de fatorial:

```
int fat(int n) {
    int res = 1;
    while(n > 0) {
        res = res * n;
        n = n - 1;
    }
    return res;
}
```

- A cada iteração o valor de n se aproxima mais de 0 e ao atingir 0 a função termina e retorna o resultado

Recursão

- Vários procedimentos computacionais podem ser descritos de forma recursiva, ou seja, que utiliza o mesmo procedimento para solução de subproblemas

- Por exemplo, a função fatorial pode ser descrita:

$$fat(n) = n * fat(n - 1)$$

Recursão

- A partir dessa definição podemos escrever a seguinte função recursiva para calcular o fatorial

```
int fat(int n) {
    if(n == 0)
        return n;
    else
        return n * fat(n - 1);
}
```

Caso base

Chamada recrusiva

As três leis da recursão

- Todo algoritmo recursivo deve obdecer a três leis:
  - Deve ter um **caso base**
  - Deve **alterar seu estado** para se aproximar do caso base
  - Deve ter uma **chamada a si mesmo**

- Para o caso da função fatorial recursiva:
  - O caso base é o fatorial de 0
  - Para se aproximar do caso base, o fatorial de n é calculado como n * fat(n - 1)
  - A chamada recursiva é feita diretamente usando fat(n - 1)

Exemplo
Série de Fibonacci

- Escreva uma função recursiva que calcule o n-ésimo valor da série de Fibonacci:

0, 1, 1, 2, 3, 5, 8, 13, ...

  - Caso base: fib(1) = 0 e fib(2) = 1
  - Método de aproximação: Fibonacci de n é calculado como a soma de fibonnaci de n – 1 e de n - 2
  - Chamada recursiva: fib(n) = fib(n - 1) + fib(n - 2)

Exemplo
Série de Fibonacci

```
int fib(int n) {
    if(n == 1)
        return 0;
    else if(n == 2)
        return 1;
    else
        return fib(n - 1) + fib(n - 2);
}
```

Caso base

Chamada recrusiva

Exemplo
Somatória de um vetor

- Escreva uma função recursiva que calcule a soma dos elementos de um vetor:
  - Caso base: soma(vet, 1) = vet[0]
  - Método de aproximação: A soma de um vetor de tamanho n é calculada como a soma do vetor na última posição com a soma de todos os elementos do vetor considerando seu tamanho como n - 1
  - Chamada recursiva: soma(vet, n) = vet[n - 1] + soma(n - 1)

Exemplo
Somatória de um vetor

```
int somaVet(int vet[], int n) {
    if(n == 1)
        return vet[n - 1];
    else
        return vet[n - 1] + somaVet(n - 1);
}
```

Caso base

Chamada recrusiva

Exercício

- O que será impresso no programa a seguir:

```
void imprime(int v[], int i, int n){
     if(i==n){
          printf("%d, ", v[i]);
     }
     else {
          imprime(v, i+1, n);
          printf("%d, ", v[i]);
     }
}

int main(){
     int vet[] = {1,2,3,4,5,6,7,8,9,10};
     imprime(vet, 0, 9);
     printf("\n");
}
```

Exercício

- Escreva uma função recursiva que calcule o número de divisores de um número n

Exercício

- Crie uma função recursiva que receba um número inteiro positivo n e calcule o somatório dos números de 1 a n.

Exercício

- A multiplicação de dois números inteiros pode ser feita através de somas sucessivas. Escreva uma função recursiva

  int multiplicaRec(int n1, int n2)

  que calcule a multiplicação de dois inteiros através de somas.

Bibliografia

- Anido, R., Notas de aula. UNICAMP, disponível em http://www.ic.unicamp.br/~ranido/mc102/, acessado em Maio de 2015.

- Mota, V. F., Notas de aula. UFMG. Disponível em https://sites.google.com/site/virginiaferm/home/disciplinas acessado em Maio de 2015.

- Deitel, P, Deitel, H. C How to Program. 6a Ed. Pearson, 2010.