MC102 - Algoritmos e Programação de Computadores

Turma Z - Segundo Semestre de 2019

Construindo um algoritmo recursivo

1. **Problema**: defina o problema
   a. Exemplo: Quero implementar a função “somar(vetor)”, que soma os elementos de um vetor de tamanho **n.**
2. **Hipótese**: Suponha que você já tem uma função que funcione para uma instância menor do problema.
   a. Exemplo: Eu já tenho e posso utilizar uma função “somar(vetor)” que soma corretamente os elementos de vetores com **até (n - 1)** elementos.
3. **Passo**: Utilize a hipótese (item 2) para resolver o problema (item 1):
   a. Exemplo: Seja `vet` o vetor de **n** elementos que eu quero somar:
      i. Retiro o primeiro elemento de `vet`, produzindo assim `vet2` (em Python: `vet2 = vet[1:]`).
      ii. Calculo a soma dos elementos de `vet2` utilizando a função `somar(vet2)` (note: eu posso fazer isso porque `vet2` tem **n - 1** elementos e minha hipótese diz que `somar` já funciona para qualquer vetor com até **n - 1** elementos.)
      iii. Retorno a soma de `vet` como sendo seu primeiro elemento somado ao retorno de “`somar(vet2)`” (em Python: `return vet[0] + somar(vet2)`).

4. **Base**: Também chamada de condição de parada. Resolva manualmente casos pequenos e aqueles que não podem ser resolvidos com o passo.

    a. Exemplo: Se eu chamar `somar(vetor)` com um vetor vazio, devo retornar o valor 0.

Exemplo de código:

```
def somar(vetor):
    if(len(vetor) == 0):
        return 0
    return vetor[0] + somar(vetor[1:])
```

Exercício 1

Considere a seguinte sequência: 1, 3, 3, 9, 27, 243

O n-ésimo elemento desta sequência é igual a multiplicação dos dois anteriores:

$$Elemento_N = Elemento_{N-1} * Elemento_{N-2}$$

Implemente uma função recursiva para calcular o n-ésimo elemento.

Exercício 2

Prova 2 - 2s2018.

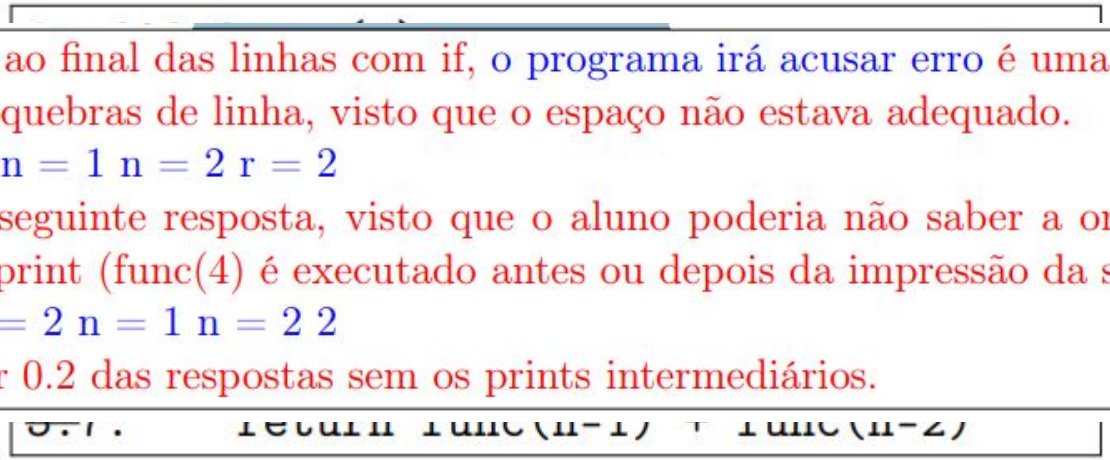
Desconsiderando os erros sintáticos (falta dois pontos após os ifs), o que o código `print("r = ", func(4))` irá imprimir?

```
1: def func (n):
2.     print("n = ", n)
3:     if n == 1
4:         return 0
5:     if n == 2
6:         return 1
~~5:~~7:     return func(n-1) + func(n-2)
```

Exercício 2

Prova 2 - 2s2018.

Desconsiderando os erros sintáticos (falta dois pontos após os ifs), o que o código `print("r = ", func(4))` irá imprimir?



Devido à falta de : ao final das linhas com if, o programa irá acusar erro é uma resposta válida.
Ignorar respeito às quebras de linha, visto que o espaço não estava adequado.
n = 4 n = 3 n = 2 n = 1 n = 2 r = 2
Aceitar também a seguinte resposta, visto que o aluno poderia não saber a ordem de avaliação dos parâmetros para o print (func(4) é executado antes ou depois da impressão da string r =???).
r = n = 4 n = 3 n = 2 n = 1 n = 2 2
Sugestão: descontar 0.2 das respostas sem os prints intermediários.

Exercício 3

a. Implemente uma função recursiva `buscar(vet, v)`, que busque o valor **v** no vetor **vet.**

b. Implemente uma função recursiva `buscarBin(vet, v)`, que busque o valor **v** no vetor ordenado **vet**, utilizando busca binária.

Exercício 4

Implemente uma função de ordenação recursiva.