UEM - UNIVERSIDADE ESTADUAL DE MARINGÁ

Fundamentos de Algoritmos

Prof. Wesley Romão

13ª Lista de Exercícios (Recursividade)

1) Dada a função X:

```
def X(n, m):
    if n == m or n == 0:
        x = 1
    else:
        x = X(n-1, m) + X(n-1, m+1)
    return x
```

a) qual o valor de X(5,3)?
b) quantas chamadas serão feitas na execução do item (a)?

2) Dada a função abaixo:

```
def X(n):
    if n >= 0 and n <= 2:
        x = n
    else:
        x = X(n-1) + X(n-2) + X(n-3)
    return x
```

a) quantas chamadas serão executadas para avaliar X(6)?
b) indique a sequencia temporal destas chamadas.

3) Sem utilizar qualquer estrutura de repetição (for, while), escreva um algoritmo que inverta os elementos de um arranjo (vetor) com N elementos, sem utilizar outro arranjo.

Exemplo:
ENTRADA: JHFCA
SAÍDA: ACFHJ

4) Escreva uma função não recursiva para as seguintes funções:

a) função recursiva

```
def fa(i):
    if i > 1:
        soma = i + fa(i - 1)
    else:
        soma = 1
    return soma
```

b) função recursiva:

```
def fb(i):
    if i == 0:        soma = 0
    elif i == 1:      soma = 1
    else:
        soma = fb(i - 1) + fb(i - 2)
    return soma
```

5) Imprimir um vetor recursivamente usando o índice.

6) Escreva um algoritmo recursivo que verifique se uma palavra é ou não palíndrome. Uma palavra palíndrome é uma que se lê da mesma forma, tanto da esquerda para a direita quanto da direita para a esquerda. Exemplo: RADAR, OMO, OVO, ARARA...

7) Escreva uma função **não recursiva** em *Python* que liste todos os pares de inteiros positivos que somados resulta em um dado número N.

Por exemplo, N=7:

7 = 6+1, 5+2, 4+3.

Não repita par algum (i.e. Não apresentar 6+1 e 1+6)

Execute para N = 7 e N = 8

8) Escreva uma função **recursiva** em *Python* que liste todos os pares de inteiros positivos que somados resulta em um dado número N. Por exemplo: 7 = 6+1, 5+2, 4+3. Não repita par algum (i.e. Não apresentar 6+1 e 1+6)

   Execute para N = 70 e N = 80

9) Escreva uma função em *Python* que receba um número real positivo e determine e devolva (em uma tupla) o número de dígitos à esquerda e à direita do ponto decimal. *Dica*: Separe ambas as partes e repetidamente divida ou multiplique por 10.

10) Dado um conjunto com N elementos, determine todos seus subconjuntos de tamanho 0 até N.

   Exemplo:

   Entrada: a, b, c, d

   Saída:

   Subconjuntos de 0 elementos: [ ]
   Subconjuntos de 1 elemento: [a] [b] [c] [d]
   Subconjuntos de 2 elementos: [a,b] [a,c] [a,d] [b,c] [b,d] [c,d]
   Subconjuntos de 3 elementos: [a,b,c] [a,b,d] [a,c,d] [b,c,d]
   Subconjuntos de 4 elementos: [a,b,c,d]

   DICAS: Lembre-se que o número de subconjuntos possíveis a partir de um conjunto com N elementos é a soma da linha N+1 do triângulo de Pascal.

   Faça um algoritmo recursivo para resolver esse problema.