Programação Estruturada - Turma A - Fabrício Olivetti
Lista de Exercícios sobre Recursão

Questão 1. Dado um vetor ordenado **v** e seu tamanho n, escreva um algoritmo recursivo para busca binária.

Questão 2. O coeficiente binomial $C(n, k)$ pode ser calculado por:

$$C(n,k) = \frac{n!}{k!(n-k)!}.$$

Mas sabemos que:

$$\begin{aligned} C(n,0) &= C(n,n) = 1 \\ C(n,k) &= C(n-1,k-1) + C(n-1,k) \end{aligned}$$

Escreva uma função recursiva para calcular C(n,k).

Questão 3. O elemento $P_{i,j}$ do triângulo de Pascal pode ser calculado da seguinte forma:

$$P_{i,j} = \begin{cases} 1, \text{se } j = 0 \text{ ou } i = j \\ P_{i-1,j-1} + P_{i-1,j} \end{cases}$$

Escreva uma função recursiva que calcula o elemento $P_{i,j}$ e escreva um programa para imprimir o triângulo de Pascal.

Responda:

1. Com qual outro algoritmo esse algoritmo se assemelha?
2. Esse algoritmo faz cálculos redundantes?

Questão 4. Para verificar se um número é primo recursivamente da seguinte forma:

$$Primo(n,d) = \begin{cases} 1, \text{se } d = 1 \\ 0, \text{se } n\%d = 0 \\ Primo(n, d-1) \end{cases}$$

Iniciando com $Primo(n, \sqrt{d})$. Escreva o código para esse algoritmo.

Questão 5. Implemente o cálculo da sequência Macho-Fêmea de Hofstadler calculada como:

$$F(n) = \begin{cases} 1, \text{se } n = 0 \\ n - M(F(n-1)) \end{cases}$$

e

$$M(n) = \begin{cases} 0, \text{se } n = 0 \\ n - F(M(n-1)) \end{cases}$$

Questão 6. Escreva o algoritmo para inverter uma string dado o algoritmo iterativo:

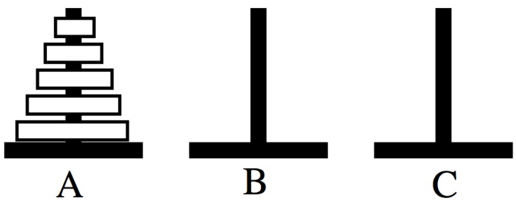
```
void inverte(s, n)
{
    int i;
    for (i=0; i<n; i++) {
        swap(s[i], s[n-1-i]);
    }
}
```

Questão 7. O algoritmo para determinar se uma string é palíndrome pode ser definida recursivamente por:

$$palindrome(s, i, j) = \begin{cases} 0, \text{se } s[i] \neq s[j] \\ 1, \text{se } i \geq j \\ palindrome(s, i+1, j-1) \end{cases}$$

Crie uma função recursiva com apenas dois parâmetros: $palindrome(s, n)$.

Questão 8. O problema da Torre de Hanoi, ilustrado na figura abaixo, é composto de 3 torres, denominadas A, B, e C e N discos. O objetivo é transportar os N discos da torre A até a torre C movendo um disco por vez e com a restrição de que um disco maior nunca deve ficar por cima do disco menor.



A solução para esse problema pode ser encontrada recursivamente se pensarmos no caso trivial: transportar apenas 1 disco de uma certa torre até a torre destino. Da mesma forma, a resposta para N discos pode ser obtida primeiro resolvendo o caso de $N-1$ discos, em uma torre auxiliar, movendo o disco restante para a torre destino, e repetindo o processo de $N-1$ discos.

Questão 9. O conjunto de todos os subconjuntos de A é denominado conjunto partes ou de potência de A, denotado por $P(A)$ ou 2^A. Exemplo:

$$A = \{1, 2, 3\} \tag{1}$$
$$P(A) = \{\{\}, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\} \tag{2}$$

Uma forma de gerar $P(A)$ é utilizando uma estrutura de sequência que guarda a sequência que será impressa a seguir. Pense nessa estrutura como uma lista ligada inversa, em que um elemento aponta para o anterior. Nesse caso podemos definir:

```
void P(v, n, seq)
{
    if (n==0) imprime(seq);
    else:
        P(v+1, n-1, seq);
        adiciona(seq, v[0]);
        P(v+1, n-1, sequ);
}
```

Escreva o algoritmo na linguagem C. Teste e mensure o tempo de execução para $n = 3, 4, 5$, qual o acrescimo de tempo entre n e $n+1$?