

Roteiro do Laboratório 02

Laboratório de Algoritmos e Estrutura de Dados II

Problema 1 Calcule o tempo (em segundos) necessário para rodar esse algoritmo:

Algoritmo demorado

```
1
2 int algoritmoDemorado(int n) {
3     int soma = 0;
4     for (int i = 0; i < n; i++) {
5         for (int j = 0; j < n; j++) {
6             for (int k = 0; k < n; k++) {
7                 for (int l = 0; l < n; l++) {
8                     soma = i + j + k + l;
9                 }
10            }
11        }
12    }
13    return soma;
14 }
```

Problema 2 Calcule a média de forma iterativa e recursiva de uma sequência V de elementos. Para isso, considere o algoritmo que gera sequências aleatórias para as quantidades:

$$n \in \{500, 1000, 1500, 2000\}$$

A forma iterativa da média é:

$$\bar{x}_n = \frac{1}{n} \sum_{i=0}^{n-1} V[i]$$

A forma recursiva pode ser calculada da seguinte forma:

$$\bar{x}_n = \begin{cases} x_0, & \text{caso } n = 1 \\ \frac{1}{n} (x_{n-1} + (n-1) * \bar{x}_{n-1}), & \text{caso contrário} \end{cases}$$

Como auxílio, logo abaixo está um gerador de sequências aleatórias e um exemplo de sua utilização.

Gerador de sequência aleatória

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <time.h>
4
5 float* gerador(size_t n) {
6     srand(time(NULL));
7     float* res = malloc(n * sizeof(int));
8     for (size_t i = 0; i < n; i++) {
9         res[i] = rand() % 1000;
10    }
11    return res;
12 }
13
14 int main() {
15     printf("Sequencia aleatoria\n");
16     size_t n = 100;
17     float *res = gerador(n);
18     for (size_t i = 0; i < n; i++) {
19         printf("res [%lu]=%f\n", i, res[i]);
20     }
21 }
```

Problema 3 Escreva um programa recursivo que recebe como entrada uma *string* (char*) e determine recursivamente se essa entrada é um palíndromo (ignore espaços, vírgulas e acentuação):

Exemplo 1:

Exp1: Ria, só tive sorte má, a metros evito sair.

Exp1 é um palíndromo.

Problema 4 Calcule recursivamente a função binomial tal como descrito abaixo:

$$\binom{n}{k} = \begin{cases} 1, & \text{if } k = n = 0. \\ \binom{n-1}{k-1} + \binom{n-1}{k}, & \text{caso contrário.} \end{cases}$$

Problema 5 Desenvolva um algoritmo contFib(n, k) que determina quantas vezes o fib(k) é avaliado durante a procura de fib(n). (Considere k < n).

Problema 6 Defina uma função que utilize como entrada um vetor de inteiros e retorne recursivamente esse mesmo vetor com os valores invertidos.

Problema 7 O cálculo da raiz quadrada pode ser feita através da aproximação:

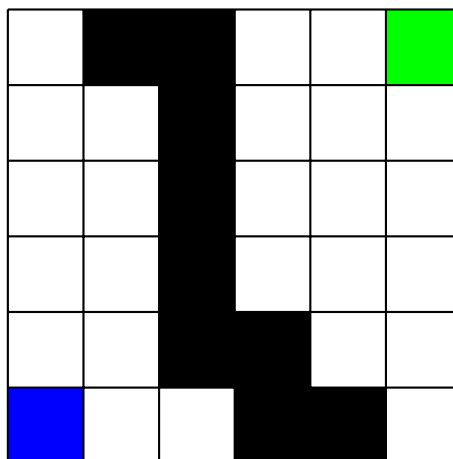
$$raiz(x, x_0, \epsilon) = \begin{cases} x_0, & \text{se } |x_0^2 - x| \leq \epsilon \\ raiz(x, \frac{x_0^2 + x}{2x_0}, \epsilon) & \end{cases}$$

Sendo que x_0 é uma aproximação inicial do valor \sqrt{x} e ϵ é um erro admissível, implemente o algoritmo e mostre o diagrama o resultado da execução de $raiz(13, 3.5, 0.001)$.

Problema 8 Depure a função Fatorial e Fatorial com cauda.

Desafio 1 Considere um quadrado $N \times N$ que consiste de valores preto e branco. Construa uma função recursiva que, a partir de uma posição inicial com branco, determine o tamanho da região quadrada.

Exemplo:

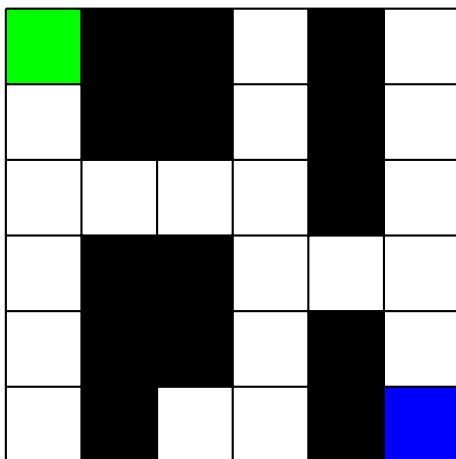


- Neste Grid, a área inicializada pelo quadrado azul possui como tamanho 12 quadrados.
- A área iniciada pelo quadrado verde é de 15 quadrados.
- A área em preto representa a barreira que os separam.

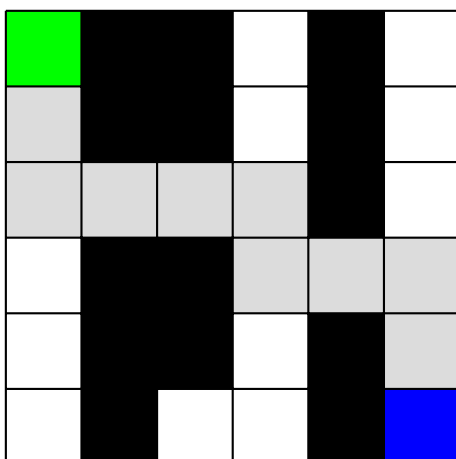
Desafio 2 Baseado na estrutura do desafio anterior, considere um grid que possua um caminho inicial e final. Determine um caminho para saída desse labirinto.

Exemplo:

Um possível caminho para o labirinto abaixo:



É esse:



A solução deve indicar quais os passos devem ser realizados para encontrar a saída.