

Aula06 - Exercícios

1. Converta sua implementação das buscas binária e sequencial de recursiva para iterativa utilizando as técnicas apresentadas na aula sobre recursividade, e compare o resultado com as versões iterativa desenvolvidas por você (na aula anterior).
2. Um problema típico em informática consiste em converter um número da sua forma decimal para a forma binária. Por exemplo, o número 10 tem a sua representação binária igual a 1010. A forma mais simples de fazer isso é dividir o número sucessivamente por 2, onde o resto da i -ésima divisão vai ser o dígito i do número binário (da direita para a esquerda). Escreva uma sub-rotina recursiva para transformar um valor inteiro não-negativo em binário. A sub-rotina deve receber um valor inteiro e enviar para a saída padrão os 0s e 1s correspondentes ao valor passado em binário.
3. Se a sub-rotina do exercício precedente não possuir recursão de cauda, transforme-a em uma que possua e em seguida escreva a sua versão iterativa.
4. Desenvolva algoritmos recursivos para os seguintes problemas e indique qual tipo de recursão está sendo utilizada em cada um dos algoritmos:

a. Cálculo de $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{N}$

b. Cálculo de $\frac{2}{4} + \frac{5}{5} + \frac{10}{6} + \frac{17}{7} + \frac{26}{8} + \dots + \frac{(n^2+1)}{(n+3)}$

5. Converta uma das funções anteriores para uma versão iterativa.
6. Realize uma análise empírica das implementações dos exercícios 4 e 5, considerando o tempo de execução. Para isso, pode-se utilizar a biblioteca `time.h` em seu programa. o funcionamento básico consiste em registrar o tempo inicial, chamar o procedimento a ser testado, e em seguida registrar o tempo final, calculando a diferença para se encontrar o tempo de execução.

Exemplo de solução para Linux

```
#include <sys/time.h>
```

```
...
```

```
/* registra a hora inicial */
```

```
struct timeval t0, t1;
```

```
gettimeofday(&t0, NULL);
```

<operacao> //funcao cujo o tempo de execucao está sendo observado

```
/*registra a hora final e calcula a diferença entre as duas */
gettimeofday(&t1, NULL);
t = (t1.tv_sec - t0.tv_sec) * 1000000; //diferença em microsegundos
t += t1.tv_usec - t0.tv_usec;
```

Para essa análise, considere que o valor de n varia entre 50 e 10000 (a cada 50). Apresente os gráficos obtidos.

7. (Desafio) Seja um labirinto descrito através de uma matriz de booleanos, onde cada posição com valor igual a verdadeiro corresponde a uma passagem livre e uma posição falsa representa uma obstrução. Escreva um algoritmo que encontre um caminho que leve uma posição inicial qualquer a uma saída do labirinto, caso exista. Uma saída é uma posição livre na borda da matriz que define o labirinto. Indique em seguida qual tipo de recursão está sendo utilizada no algoritmo desenvolvido.

8. (Desafio) Uma imagem discreta de largura w e altura h , pode ser representada em um computador através de uma matriz $I[i, j]$, de ordem $w \times h$, que armazena em cada posição um número inteiro entre **0** e **255**, o qual especifica uma certa cor em uma paleta de cores. Em pacotes de pintura interativos, é muito comum haver uma operação que efetua o preenchimento de certa área de uma imagem contendo uma cor $c_{anterior}$ com uma nova cor c_{nova} . Esta operação pode ser realizada de forma simples através de um método denominado *Boundary-fill*.

O procedimento em questão recebe como entrada um ponto no interior da região especificado por índices (x, y) e a cor de preenchimento c_{nova} . O algoritmo inicialmente detecta a cor $c_{anterior}$ no ponto (x, y) e começa pintando tal posição com a cor c_{nova} caso $c_{nova} \neq c_{anterior}$. O processo é repetido recursivamente para os vizinhos acima $I[x, y+1]$, abaixo $I[x, y-1]$, à esquerda $I[x-1, y]$ e à direita $I[x+1, y]$ desde que estejam dentro da imagem e possuam cor igual a $c_{anterior}$, isto é, igual a cor a ser substituída.

Escreva uma sub-rotina que implemente tal algoritmo a fim de obter resultado conforme a ilustração abaixo. Por fim, indique qual tipo de recursão foi utilizada na implementação.

