

## Lista de Exercícios 1

Rodrigo de Souza

Abril 2015

1. Explique o que significa dizer que um algoritmo tem complexidade de pior caso  $\mathcal{O}(n \lg n)$ , onde  $n$  é o tamanho da entrada.
2. Considere a operação de multiplicação de matrizes, realizada de forma usual (multiplicando-se as linhas da primeira matriz pelas colunas da segunda). Considere uma matriz quadrada  $A$  de dimensão  $n \times n$ . Relativamente a um algoritmo para realizar a operação  $A \cdot A$ , estime o tamanho da entrada e sua complexidade.
3. Ainda sobre o problema de multiplicação de matrizes, considere o produto  $(A \cdot B) \cdot x$ , onde  $A$  e  $B$  são matrizes quadradas  $n \times n$  e  $x$  é um vetor de dimensão  $n$ .
  - (a) Qual é o número de operações desse produto?
  - (b) Há alguma maneira (simples) de realizar esse produto de forma mais eficiente?
4. Escreva um programa recursivo que recebe um vetor de inteiros  $A[1..n]$  e um inteiro  $k$ , e devolve um posição  $j$  tal que  $A[j] = k$ . Faça um comentário explicando o que seu programa faz. Precise no comentário a posição devolvida pelo seu programa (primeira, última, etc.) e o que o programa devolve caso tal posição não exista. Usando uma prova por indução, estime um limitante superior para o número de chamadas recursivas do seu programa.
5. A função recursiva seguinte promete resolver o problema anterior (considerando que  $A$  é um vetor indexado de 0 a  $n - 1$ ):

```
int busca(int A[], int n, int k) {  
    if(n < 1)  
        return -1;  
    int l, r, q;  
    q = n / 2;  
    l = busca(A, q, k);  
    r = busca(A + q, q, k);  
    return l > r ? l : r;  
}
```

- (a) Explique a intenção do programador.

- (b) Simule a execução do programa para alguns valores de  $A$ ,  $n$  e  $k$ .
  - (c) Critique o programa. Corrija-o se necessário.
  - (d) Usando uma prova por indução, estime um limitante superior para o número de chamadas recursivas. Que paradigma de projeto de algoritmos seu programa segue?
6. Escreva um programa recursivo que recebe um inteiro positivo  $n$  e calcula  $2^n$ . Usando uma prova por indução, estime um limitante superior para o número de chamadas recursivas do seu algoritmo.
  7. Escreva um programa que recebe um vetor de inteiros  $A[1..n]$  e decide (responde SIM ou NÃO) se há duas posições em  $A$  contendo o mesmo valor. Calcule a complexidade do seu algoritmo.
  8. Escreva um programa que recebe dois vetores ordenados  $A[1..m]$  e  $B[1..n]$ , e constrói um vetor  $C[1..n+m]$  contendo precisamente os valores em  $A$  e  $B$  em ordem crescente. Analise a complexidade do seu algoritmo.
  9. Escreva um programa que recebe um vetor  $A[1..n]$  e rotaciona-o para a direita, ou seja: troca os elementos de  $A$  de lugar de forma que o vetor resultante  $B$  satisfaça  $B[k] = A[k-1]$  para  $2 \leq k \leq n$  e  $B[1] = A[n]$ . Analise a complexidade do seu algoritmo.
  10. Considere o seguinte trecho de código em C, que promete rotacionar um vetor  $A$  para a direita:

```
A--;  
A[0] = A[n];
```

- (a) Explique a intenção do programador.
  - (b) Critique o programa.
11. Escreva um programa que receba uma matriz quadrada  $M[1..n][1..n]$  de inteiros e testa se  $M$  é um quadrado mágico:  $M$  não tem duas posições com o mesmo número, e a soma de cada linha, cada coluna e cada diagonal é o mesmo inteiro.