# Análise de Algoritmos – Lista II

**Ex. 1 —** (Aquecimento) Determine a complexidade de tempo do algoritmo de ordenação a seguir.

```
ORDENA(V,n):
para i de 1 a n:
    para j de 1 a n:
        se V[j]<V[i]:
            tmp = V[i]
            V[i]=V[j]
            V[j]=tmp
```

**Ex. 2 —** (Aquecimento) Explique o que faz o estranho algoritmo a seguir.

```
bt(V,a,b,x):
p=  (a-b)/3
q= 2(a-b)/3
se x==V[p]: retorne p
se x==V[q]: retorne q
se x<V[p]:  retorne bt(V,a,p-1,x)
se x<V[q]:  retorne bt(V,p+1,q-1,x)
senao:      retorne bt(V,q+1,b,x)
```

Como seria a complexidade de tempo? Vale a pena usá-lo?

**Ex. 3 —** Mostre como remover elementos duplicados de um vetor em tempo $\mathcal{O}(n \log n)$. O novo vetor pode ser menor que o original, claro.

**Ex. 4 —** Prove que qualquer algoritmo para encontrar um elemento em um vetor ordenado usando comparações terá complexidade de tempo $\Omega(\log n)$.

**Ex. 5 —** Descreva um algoritmo com consumo de tempo $\mathcal{O}(n \log n)$ que tenha como entrada:

- conjunto $S$ contendo $n$ números inteiros
- um número inteiro $x$,

e determine se existem ou não dois elementos em $S$ cuja soma é exatamente $x$.

**Ex. 6 —** Dado um vetor $A[1..n]$ de inteiros, chamamos de uma inversão de $A$ um par de índices $(i, j)$ do vetor $A$ tal que $i < j$ e $A[i] > A[j]$. Escreva um algoritmo `conta-inversoes(A,n)` que recebe um vetor $A[1..n]$ com inteiros todos distintos e devolve o numero de inversoes de $A$. Seu algoritmo deve consumir tempo $\mathcal{O}(n \log n)$ Explique sucintamente porque seu algoritmo esta correto e porque ele tem a complexidade de de tempo pedida.

**Ex. 7 —** Construa um algoritmo que receba um vetor de $n$ elementos e retorne o tamanho do maior subvetor palíndromo. Analise a complexidade de tempo de seu algoritmo[1].

[1] É possível fazer em $\mathcal{O}(n^2)$.

**Ex. 8 —** O determinante de matrizes quadradas pode ser calculado de diversas maneiras. Entre elas temos:

a) a expansão de Laplace (menores, cofatores, etc)

b) deixar a matriz na forma triangular e calcular o produtório da diagonal

Calcule a complexidade de tempo dos dois métodos. Se o método for normalmente descrito como algoritmo recursivo, mostre a recorrência que dá a complexidade.