

### Lista de Exercícios 1 – Análise de Complexidade

#### Questão 1

Considere o algoritmo iterativo mostrado abaixo que encontra o maior e o menor elemento em um vetor  $A[1..n]$ . Considere ainda que os  $n$  elementos estão distribuídos aleatoriamente no vetor.

```
MaxMin(int A[1..n]) {  
    max = A[1];  
    min = A[1];  
    for(i=2; i<=n; i++)  
        if (A[i]>max) max = A[i];  
        else if(A[i]<min) min = A[i];  
    imprime(min,max)  
}
```

- a) Utilizando invariantes de loop, mostre que esse algoritmo funciona.
- b) Qual é a função de complexidade do número de comparações de elementos no melhor e pior caso?
- c) Utilizando análise probabilística, compute o número de comparações de elementos do vetor que serão realizadas no caso médio.
- d) Implemente um **algoritmo recursivo** *MaxminRec* usando o paradigma “dividir para conquistar” para resolver esse mesmo problema. (dica: modifique o mergesort). A complexidade do seu algoritmo deve ser **inferior a  $n \cdot \log(n)$** .
- e) Qual é a complexidade do seu algoritmo? Para isso, **determine e resolva** a sua equação de recorrência.

#### Questão 2

Sejam  $f(n)$ ,  $g(n)$  duas funções assintóticas positivas. Prove que as afirmativas abaixo são verdadeiras ou falsas, usando para isso as definições das notações assintóticas ou contra exemplos.

- a) A relação  $\Theta$  é simétrica, ou seja  $g(n) = \Theta(f(n))$  se e somente se  $f(n) = \Theta(g(n))$ .
- b)  $f(n) = O(f(n/2))$
- c)  $f(n) = O(f(n)/2)$  (obs: note que a equação é diferente da letra b)
- d) Considerando que  $f(n) = o(g(n))$  então  $f(n) + g(n) = \Theta(g(n))$
- e)  $(n+a)^b = \Theta(n^b)$

#### Questão 3

Encontre um **limite assintótico firme** para as equações de recorrência abaixo usando os métodos indicados:

- a)  $T(n) = T(n/2) + 1$ , usando o método da expansão de termos.
- b)  $T(n) = T(n-1) + n^2$ , usando o método da substituição.
- c)  $T(n) = 2T(n/2) + n^3$ , usando o teorema mestre.