IPRJ – PROJETO E ANÁLISE DE ALGORITMOS

LISTA DE EXERCÍCIOS 01

1) Supondo que estamos comparando duas implementações de um algoritmo em um mesmo computador. Para entradas de tamanho *n*, a implementação A é executada em $8n^2$ etapas, enquanto a implementação B é executada em $64n \log n$ etapas. Para que valores de *n* a implementação A supera a implementação B?

2) As funções $\log n$ e $\log n^2$ possuem a mesma ordem de complexidade? Justifique sua resposta.

3) Dada a função $T(n) = 64n^3 + n \log n + 128n$, responda verdadeiro ou falso para às afirmações abaixo.
   a) $T(n) = O(n \log n)$
   b) $T(n) = \Omega(\log n)$
   c) $T(n) = \Theta(n^3)$
   d) $T(n) = O(n^3)$
   e) $T(n) \neq \Omega(n)$
   f) $T(n) = O(1)$
   g) $T(n) = \Omega(1)$
   h) $T(n) \neq O(n!)$
   i) $T(n) = \Theta(n \log n)$
   j) $T(n) = O(2^n)$

4) Escreva um algoritmo que, dado um conjunto *S* de *n* inteiros e outro inteiro *x*, determina se existe ou não dois elementos de *S* cuja soma é exatamente *x*. Em seguida, análise a complexidade deste algoritmo.

5) Análise a complexidade dos algoritmos abaixo:

a)
```
float func1(int n, float A[], float x)
{
    int  k;
    float y = 0.0;
    for (k = n; k >= 0; k--)
    {
        y = A[k] + y * x;
    }
    return  y;
}
```

b)
```
int func2(int n)
{
    int  i, j, x, soma = 0;
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < n; j++)
        {
            for (x = 0; x < n; x++)
            {
                soma += n;
            }
        }
    }
    return  soma;
}
```

c)
```
void func3(int* A, int n)
{
    int i, j, aux;
    for( j = 2; j <= n; j++){
        aux = A[j];
        i = j - 1;
        while (i > 0 && A[i] > aux){
            A[i + 1] = A[i];
            i = i -1;
        }
        A[i + 1] = aux;
    }
}
```

6) Dadas *n* variáveis booleanas, escreva um algoritmo que gere todas as combinações possíveis. Por exemplo, para três variáveis deverá ser gerado: 000 – 001 – 010 – 011 – 100 – 101 – 110 – 111. Em seguida, análise a complexidade deste algoritmo.