

Centro de Ciências Exatas e da Tecnologia  
CIC4003A - Complexidade de Algoritmos  
Prof. Ricardo Dorneles  
Prova 1 - 04/10/22

1)(2 pontos) Resolva as seguintes equações de recorrência, supondo  $T(1)=1$ :

1.  $T(n) = 3T(n/2) + 1$
2.  $T(n) = 10T(n/4) + n$

2)(2 pontos) Para o subalgoritmo recursivo abaixo, determine a equação de recorrência que representa a complexidade do mesmo e a equação fechada correspondente. Verifique sua solução por indução matemática.

```
function pl(int n)
{
    if (n>1)
    {
        pl(n/2);
        for (i=1;i<=n*n;i++)
            soma = soma + i;      // <- considere operacao fundamental
        for (i=1;i<=n;i++)
            soma = soma+i*i;      //<- aqui também
    }
}
```

3)(2 pontos) Um programa com complexidade  $n^3$  atende as necessidades de uma empresa que processa dados de 800 funcionários em duas horas.

1. Se o número de funcionários reduzir para a metade, qual será o tempo de execução deste programa?
2. Se o computador for trocado por um outro 4 vezes mais rápido, qual o número máximo de funcionários que este programa processa em 4 horas?

4)(2 pontos) Os laços abaixo formam a estrutura para preencher a matriz no problema da ordem dos produtos matriciais. Calcule sua complexidade:

```
for (i=1;i<N;i++)
  for (j=0;j<N-i;j++)
    for (k=i-2;k>=0;k--)
      c=c+1; // operação fundamental
```

5)(2 pontos) Calcule a complexidade do trecho de código a seguir, que triangulariza uma matriz quadrada:

```
for (i=0;i<n;i++)
{
  for (k=n-1;k>=i;k--)
    m[i][k]/=m[i][i]; <- op fund
  for (k=i+1;k<n;k++)
    for (j=n-1;j>=i;j--)
      m[k][j]=m[k][j]/m[k][i]-m[i][j]; <- op fund.
}
```