

Exercícios Aula 04

<https://daniloeler.github.io/teaching/PAA2020/index.html>

1) Use o teorema mestre para calcular o limite assintótico das seguintes recorrências. Se não for possível aplicar o teorema, somente indique que não é possível.

$$(a) T(n) = 4T(n/2) + n$$

$$(b) T(n) = 4T(n/2) + n^2$$

$$(c) T(n) = 4T(n/2) + n^3$$

$$(d) T(n) = 7T(n/2) + n^2$$

$$(e) T(n) = T(n/2) + 1$$

$$(f) T(n) = 3T(n/5) + n \log n$$

$$(g) T(n) = 3T(n/4) + n \log n$$

2) Calcule a complexidade do subprograma abaixo. O método subArray retorna um array a partir dos índices indicados por parâmetro.

```
public void subAlg03(ArrayList args){
    int n=args.size();
    if (n<=1){
        exibir("veio aqui");
    }
    else{
        for (int i=0; i< n*n; i++){
            exibir("passou aqui");
        }
        subAlg03(args.subArray(0, (int) (n/2)));
        subAlg03(args.subArray((int) (n/2)+1,n - 1));
        subAlg03(args.subArray(0, (int) (n/2)));
        subAlg03(args.subArray((int) (n/2)+1,n - 1));
    }
}
```

3) Utilize o teorema mestre para calcular a complexidade de tempo do algoritmo abaixo.

```
Algoritmo Pesquisa(vetor)
  if vetor.size() ≤ 1 then
    inspecione elemento;
  else
    inspecione cada elemento recebido (vetor);
    Pesquisa(vetor.subLista(1, vetor.size() / 3));
  end if
end.
```

4) Qual é a complexidade de tempo para o **melhor** eu o **pio**r caso do subprograma **Principal** listado abaixo?

```
void Principal(vetor X, inteiro n, elem){
  subPrograma01(X, n);
  inteiro p = subPrograma02(X, 1, n, elem);
}

void subPrograma01(vetor X, inteiro n) {
  para (k = 1; k < n; k++) {
    y = X[k];
    para (i = k-1; i ≥ 0 && X[i] > y; i--)
      X[i+1] = X[i];
    X[i+1] = y;
  }
}

inteiro subPrograma02 (vetor X, inteiro ini, inteiro n, inteiro elem){
  i = ( ini + n ) / 2;
  se (X[i] == elem) retorna i;

  se (ini == n) retorna -1;

  se (X[i] < elem)
    subPrograma02(X, i+1, n, elem);
  senão
    subPrograma02(X, ini, i-1, elem);
  fimse
}
```