# Estruturas de Dados II (DEIN0083) 2022.1
## Curso de Ciência da Computação
## Atividade Avaliativa (30% da 1ª nota)

Prof. João Dallyson Sousa de Almeida **Data**: 09/05/2022

**Aluno**: _____ **Matrícula**: | | | | | | | | | | | | | |

**Regras durante a prova:**

- É vetada: cópia de respostas dos colegas. A não observância de algum dos itens acima acarretará a anulação da prova.
- Após a avaliação, você poderá ser selecionado para uma entrevista para verificar a propriedade de suas respostas.

I. (2.0pt) Forneça as complexidades para os algoritmos seguir. Você deve primeiro escrever uma função matemática F(n) que conta exatamente quantas vezes a linha que contém a variável count é executada. Nos algoritmos recursivos você deve fornecer a relação de recorrência para F(n). Em seguida mostre a complexidade de cada algoritmo apresentando a ordem de crescimento (Big-O). Descreva a solução apresentada.

Figura 1: A

```
int test(int n)
{
  int count = 0;
  for (int i = n; i > 0; i /= 2)
     for (int j = 0; j < i; j++)
        count += 1;
  return count;
}
```

Figura 2: B

```
int func1 (int n) {
    int count = 0;
    int i, j;
    for (int i = 0; i < n; i++)
        count++;
    for (int j = 0; j < (n * n); j++)
        count++;
    return count;
}
```

Figura 3: C

```
int func3 (int n) {
    int count = 0;
    int i, j;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < (n * n); j++) {
            for (int i = 0; i < n; i=i * 2) {
                count++;
            }
        }
    }
    return count;
}
```

Figura 4: D

```
void function (int n) {
    int i, j, k, soma = 0;

    for (i=n/2; i<=n; i++)
        for(j=1; j<=n; j=3*j)
            for(k=1; k<=n; k=k*3)
                soma++;
}
```

II. (2.0pt) Apresente e demonstre o resultado da análise assintótica para as recorrências a seguir:

(A) $T(n) = 6T(n/3) + n^2 logn$ (B) $T(n) = 5T(n/5) + \sqrt{n}$
(C) $T(n) = 7T(n/7) + n/3$ (D) $T(n) = 3T(n/9) + n^{0,62}$

III. (4.0pt)Considere a seguinte lista de números inteiros [13, 21, D1, D2, D3, D4] na qual D1, D2, D3 e D4 são, respectivamente, os 4 últimos dígitos da sua matricula, exemplo: 201403[6][0][4][3] (D1=6, D2=0, D3=4 e D4=3). Responda as questões a seguir utilizando os algoritmos de ordenação estudado.

- (a) Mostre o estado da lista após as 3 iterações completadas do loop mais externo do InsertSort. Mostre a lista resultante após cada iteração.

- (B) Mostre o estado da lista após as 3 primeiras execuções do método Particiona do QuickSort. Mostre a lista resultante após cada iteração. Utilize o item mais à direita como o pivô.
- (C) Mostre o estado dos vetores auxiliares (B e C) após a inserção dos quatro primeiros itens no vetor auxiliar. Mostre a lista resultante após cada iteração.
- (D) Mostre a lista resultante após a construção do MinHeap (Heap Mínimo). Mostre o estado da lista após as 3 primeiras iterações do HeapSort. Obs: neste caso, os itens serão ordenados em ordem decrescente.

IV. (2.0pt) Escreva um algoritmo para retornar o I itens mais próximos de um valor V, para um vetor qualquer de elementos inteiros. V pode estar ou não presente no vetor. Sua solução deve ser O(nlogn).
Ex:
Entrada: [5, 3, 8, 7, 1, 4, 5]
Quantidade de itens mais próximos: 3
V = 6
Saída: 5, 7, 4
Nesse exemplo poderia ser retornado 4 ou 8. Nesses casos, priorize o elemento menor que V.