Universidade Federal do Maranhão
Departamento de Informática

Estruturas de Dados II (DEIN0083) 2021.1
Curso de Ciência da Computação
Atividade Avaliativa (30% da 1ª nota)

Prof. João Dallyson Sousa de Almeida **Data**: 29/11/2021

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 func1 (int n) {
  int count = 0;
  for (int i = 0; i < n; i++)
    count += 1;
  for (int j = 1; j < n; j++)
    count += 1;
  return count;
}
```

Figura 2: B

```
int func2(int n) {
  int count = 0;
  for (int i = 1; i <= n; i *= 3)
    for (int j = 0; j < n*n; j++)
      count += 1;
  for (int k = 0; k < n; k++)
    count += 1;
  return count;
}
```

Figura 3: C

```
int func3(int n) {
  int count = 0;
  for (int i = 0; i < n; i++)
    count += 1;
  if (n > 0)
    return count + func3(n-1);
  else
    return 0;
}
```

Figura 4: D

```
int func4(int n) {
  int count = 0;
  for (int i = 0; i < n; i++)
    count += 1;
  if (n > 1)
    return func4(n/3) + func4(n/3) + count;
  else
    return 0;
}
```

II. (1.0pt) Considere 2 softwares X e Y de complexidade O(nlogn) e O(n), respectivamente, que gastam $TX(n) = c_x n \log_{10} n$ e $TY(n) = c_y n$ milissegundo para processar n dados. Durante os testes, o tempo médio para processar $n = 10^4$ itens de dados com o software X e Y é, respectivamente, 200 milissegundos e 1000 milissegundos. Calcule as condições exatas em que um software realmente supera o outro e recomende a melhor escolha se tiver até $n = 10^9$ itens para serem processados.

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

(A) $T(n) = 3T(n/9) + \sqrt{n}$ (B) $T(n) = 5T(n/3) + (5/6)n + 2$
(C) $T(n) = 2T(n/4) + n^{0.6}$ (D) $T(n) = 16T(n/4) + n$

IV. (4.0pt)Considere a seguinte lista de números inteiros [D1, D2, D3, D4, 8, 7] 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 e apresente a estratégia utilizada na definição do 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 MaxHeap. Mostre o estado da lista após as 3 primeiras iterações do HeapSort.

V. (1.0pt) Escreva um algoritmo para reorganizar os elementos de um dado vetor com n números reais de modo que todos os seus elementos negativos precedam todos os elementos positivos. O algoritmo proposto deve ser eficiente em termos de tempo e espaço.