

**Dica:** Antes de fazer esses exercícios, leia o seguinte texto:

- [https://www.ime.usp.br/~pf/analise\\_de\\_algoritmos/aulas/0h.html](https://www.ime.usp.br/~pf/analise_de_algoritmos/aulas/0h.html)

## Noções de Análise de Algoritmos

- (2.5 points) Para cada uma das afirmações abaixo, prove se é verdadeiro ou falso, justificando formalmente (usando definições, manipulações algébricas e implicações se for preciso). **Atenção:** Está proibido usar limites para resolver esta questão.
  - $10n^2 + 200n + 500/n = O(n^2)$
  - $\lg(100n^3 + 200n + 300)^2 = O(\lg n)$
  - $2n^2 - 20n - 50 = \Omega(2n)$
  - Seja  $C(n, k)$  o número de combinações de  $n$  objetos tomados  $k$  a  $k$ . É verdade que  $C(n, 2) = O(n^2)$ ? É verdade que  $C(n, 3) = O(n^3)$ ?
- (2.5 points) Determine a complexidade de pior caso do algoritmo a seguir:

---

**Algoritmo 1** Função F

---

```
1: Função F(int L[ ], int n)
2:   s ← 0
3:   para i ← 0 até n – 2 faça
4:     para j ← i + 1 até n – 1 faça
5:       if L[i] > L[j] then
6:         s ← s + 1
7:       fim if
8:     fim para
9:   fim para
10:  retorne s
11: fim Função
```

---

- (2.5 points) Faça um algoritmo que verifique se os elementos de um vetor estão ordenados em ordem crescente. Qual a complexidade de pior caso e melhor caso do seu algoritmo? Prove que suas respostas estão corretas. **Atenção:** Note que eu não estou pedindo para ordenar o vetor.
- (2.5 points) A sequência de Fibonacci é uma sequência de elementos  $f_0, f_1, \dots, f_n$ , definida do seguinte modo:

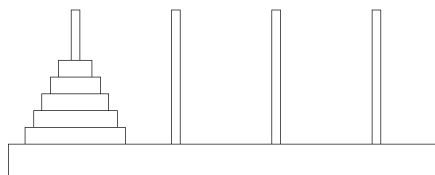
$$f_j = \begin{cases} j, & \text{se } 0 \leq n \leq 1; \\ f_{j-1} + f_{j-2}, & \text{se } j > 1. \end{cases}$$

Elaborar um algoritmo iterativo (não recursivo), para determinar o elemento  $f_n$  da sequência, cuja complexidade seja linear em  $n$  e prove este fato.

5. (1 point) (**Bônus**) Considere a seguinte generalização do problema Torre de Hanói. O problema agora consiste em  $n$  discos de tamanhos distintos e quatro pinos, respectivamente, o de origem, o de destino e dois pinos de trabalho (auxiliares). De resto, o problema é como no caso de três pinos. Isto é, de início, os discos se encontram todos no pino de origem, em ordem decrescente de tamanho, de baixo para cima. O objetivo é empilhar todos os discos no pino-destino, satisfazendo às condições:

- (i) apenas um disco pode ser movido de cada vez;
- (ii) qualquer disco não pode ser jamais colocado sobre outro de tamanho menor.

Escrever um algoritmo recursivo para resolver essa generalização. O seu programa deve ser escrito em C++ e deve imprimir na tela a sequência de movimentos que resolve esse problema para uma entrada  $n$ , onde  $n$  é o número de discos.



---

**Submissão de Respostas para a Atividade**

Para quem conhece **Latex**, pode fazer a atividade em latex. Estou enviando o arquivo .tex junto com a atividade para quem quiser.

Para quem quiser fazer em papel: Resolva as questões usando papel e caneta, **em ordem**. Logo após, tire fotos das respostas, com atenção aos seguintes detalhes:

1. LEGIBILIDADE: Suas respostas devem ser legíveis no papel e também nas fotos tiradas ao final. Verifique se suas fotos não ficaram borradas. Para facilitar, tire uma foto para cada questão submetida. Certifique-se de que você tenha escrito um cabeçalho com seu nome e matrícula na resposta da primeira questão.
2. Formato: **PDF**. Utilize a ferramenta de sua escolha para gerar um arquivo .PDF com as fotos de suas respostas na ordem em que os itens foram pedidos.
3. SUBMISSÃO: Via Moodle, faça upload do arquivo .PDF com suas respostas na seção da respectiva atividade no Moodle.
4. REQUISITOS: Você é responsável por verificar os requisitos de submissão e que o **upload funcionou corretamente**. Após submeter suas respostas no Moodle, verifique se consegue efetuar o download do arquivo e abri-lo corretamente. **Se você não verificar e ao final o arquivo não tiver sido enviado corretamente, sua nota na atividade não será contabilizada.** Não envie a solução da tarefa por email, pois ela não será considerada.