

## Proposta — Entrega: 10/02/2026 e 12/02/2026

### I. Problemas

A partir dos problemas listados, abaixo, escolher um tópico e realizar o que se pede na Seção II - Instruções Gerais.

#### Programação Dinâmica:

- Subsequência Comum Máxima (LCS):** Dadas duas sequências, encontre o comprimento da maior subsequência comum entre elas.
- Corte de Hastes (Rod Cutting):** Dada uma haste de comprimento  $n$  e uma tabela de preços para cada tamanho, determine a forma de cortar a haste para maximizar o lucro.
- Edit Distance (Levenshtein):** Calcule o número mínimo de operações (inserção, remoção e substituição) para transformar uma string em outra.
- Multiplicação de Cadeia de Matrizes:** Determine a ordem de multiplicação de uma sequência de matrizes que minimize o custo total de operações.
- Longest Increasing Subsequence (LIS):** Dado um array de inteiros, encontre o comprimento da maior subsequência crescente.

#### Algoritmos Gulosos:

- Seleção de Atividades:** Selecione o maior conjunto possível de atividades que podem ser realizadas sem que seus horários se sobreponham.
- Códigos de Huffman:** Construa uma árvore de codificação binária que minimize o comprimento médio da mensagem com base na frequência dos caracteres.
- Árvores Geradoras Mínimas (Prim/Kruskal):** Encontre o subconjunto de arestas de custo mínimo que conecta todos os vértices de um grafo sem formar ciclos.
- Algoritmo de Dijkstra:** Determine o caminho de menor custo de um único vértice de origem para todos os outros vértices em um grafo com pesos não negativos.
- Mochila Fracionária:** Maximize o valor total na mochila permitindo levar partes dos itens e demonstre a diferença para a versão inteira 0/1.

### II. Instruções Gerais

Cada grupo deverá escolher um dos problemas listados e desenvolver um seminário contendo os seguintes elementos obrigatórios:

- Contexto histórico:** Explique quando e por que o algoritmo foi criado. Mencione os autores e publicações originais.
- Descrição do problema:** Apresente de forma clara o enunciado do problema e suas aplicações práticas.
- Descrição do algoritmo:** Detalhe como o algoritmo funciona passo a passo, com exemplos ilustrativos.
- Complexidade assintótica:** Faça a análise da complexidade de tempo e espaço do algoritmo.
- Demonstração de corretude:** Explique por que o algoritmo resolve corretamente o problema para qualquer entrada válida.
- Implementação:** Entregue um código funcional que resolva o problema proposto.
- Avaliação empírica:** Gere um gráfico comparando o tempo de execução real (medido experimentalmente) com a complexidade teórica esperada.

8. **Perguntas no estilo múltipla escolha:** Cada grupo deverá elaborar 02 perguntas com 04 alternativas (de A a D), sobre o tema do seu trabalho, juntamente com o gabarito. Essas perguntas comporão um questionário a ser respondido por todos os discentes ao final das apresentações.

### III. Roteiro

Cada grupo deverá:

1. Elaborar uma solução para o problema identificado, por meio da aplicação de programação dinâmica, algoritmos gulosos e seus conceitos.
2. Realizar uma apresentação demonstrando as etapas de elaboração do trabalho e os resultados obtidos.
3. Escrever e submeter um relatório, seguindo o modelo da SBC para artigos.

### IV. Informações Gerais

1. Este trabalho equivale a **50% da nota 03**.
  - Sendo que 80% está destinado aos seminários e 20% a um questionário a ser respondido pelos discentes, relacionado aos temas dos trabalhos.
2. Composição das equipes: 03 integrantes no mínimo e 03, no máximo.
3. A ordem de apresentação será por sorteio do tópico, no dia da apresentação.
4. Tempo de apresentação: mínimo de 10 e máximo de 15 minutos.
  - No dia da apresentação, recomenda-se que os grupos se preparem para garantir um melhor aproveitamento do tempo.
  - Não é obrigatória a participação de todos os integrantes na apresentação. Contudo, todos devem estar presentes no momento da apresentação para eventuais questionamentos realizados.
  - Integrantes ausentes por motivos não justificáveis no momento da apresentação terão sua nota reduzida em 20% da nota recebida pelo grupo.
5. Um mesmo tópico só poderá ser escolhido por, no máximo, **02** grupos.
6. Tecnologias:
  - Linguagem de programação à escolha do grupo.
7. A divisão das tarefas no grupo fica a critério dos seus integrantes.
  - Integrantes ociosos [sem nenhuma ou mínima contribuição] terão sua nota reduzida em 85% da nota que o grupo receber.
  - No mínimo, 02 pessoas devem ficar responsáveis pelo desenvolvimento do código.
  - Sugestão: 02 desenvolvedores, 01 e produtor da apresentação.
8. Uso de ferramentas inteligentes será permitido:
  - na solução para resolução de problemas **pontuais**, que sozinho, o grupo não conseguiu resolver;
  - na apresentação de slides, somente para revisão, e que deverá ser validada pelos autores.
9. É permitido às equipes, com o mesmo problema, trocarem ideias entre si, todavia a implementação deve ser particular de cada grupo.
10. Embora não seja obrigatório, é altamente recomendável que cada equipe armazene os artefatos de código em uma plataforma de versionamento como Github, Gitlab, Bitbucket, etc.

### V. Cronograma

1. 05/02 até às 23:59 — Prazo para definição das equipes e escolha dos problemas na Plataforma Turing.
2. 10/12 e 12/02 a partir das 08:10 — Apresentação dos trabalhos em sala de aula.

3. 10/12 até às **07:59** — Submissão dos trabalhos de **todos** os grupos na Plataforma Turing.

## VI. Entregáveis

---

Um arquivo do tipo .zip (max. 50Mb) contendo:

1. Apresentação de slides **em PDF** seguindo modelo de slides do INF (novo), disponível no Google Slides ou [Modelos do Overleaf](#); nomeado como: “apresentacao-grupo-X.pdf”.
2. Relatório, **em PDF**, de no máximo 10 páginas, exceto referências bibliográficas. Segundo o modelo da Sociedade Brasileira de Computação.
3. Arquivo .TXT com as perguntas elaboradas pelo grupo
4. Códigos-fonte ou link do repositório público no qual o projeto foi armazenado;
5. O zip deverá estar em conformidade com a seguinte estrutura:

```
.trabalho-apo-grupo-x/          # nome da pasta do projeto
└── apresentacao-grupo-X.pdf    # apresentação dos slides em PDF
└── perguntas-grupo-X.txt      # perguntas elaboradas pelo grupo
└── README.md                   # README.md, se houver
└── relatorio-grupo-X.pdf      # relatório em PDF
└── codigo/                      # diretório com os códigos
    └── graficos/                # diretório com os gráficos e arquivos geradores
```

## VII. Relatório

---

1. O relatório poderá ser escrito em português ou inglês.
2. O relatório deverá possuir, no mínimo:
  - a. Resumo;
  - b. Introdução (contextualização, descrição do problema);
  - c. Materiais e Métodos (descrição dos algoritmos, ferramentas e justificativa de escolha);
  - d. Resultados e Discussão (complexidade assintótica, demonstração de corretude, avaliação);
  - e. Considerações Finais
  - f. Referências.
3. O relatório poderá conter imagens das soluções ou visualizações, bem como pseudocódigo;
4. O relatório deve possuir no mínimo 05 páginas e no máximo 10 páginas, excluindo-se as referências.

## VIII. Avaliação

---

Os trabalhos serão avaliados considerando os seguintes aspectos:

1. O desenvolvimento/implementação dos problemas propostos usando os conceitos aprendidos ao longo da disciplina.
2. O rigor técnico realizado em cada etapa do projeto.
3. A qualidade da apresentação (oral e visual) e do relatório.
4. A integridade do relatório (se todas as seções solicitadas foram totalmente descritas no texto).
5. A criatividade demonstrada ao resolver os problemas enfrentados pelo grupo.

**O descumprimento das observações abaixo implicará penalização na nota (5% por item).**

1. Apresentação de slides fora do modelo do INF.
2. Relatório fora do modelo da SBC.
3. Arquivos da apresentação e do relatório em formatos diferentes de PDF.

4. Tempo de apresentação inferior ou superior ao tempo especificado na Seção Informações Gerais.
5. Nomenclatura do .zip, arquivos e diretórios fora da especificação definida na Seção Entregáveis.