



Simulador de Missões Espaciais Inteligente

AstroSim

Estruturas de Dados II: O Mestre dos Algoritmos!



**UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO**

Kensley Alves de Oliveira
32411BSI037

Simulador de Missões Espaciais Inteligente AstroSim

Monte Carmelo
2025



RESUMO

AstroSim é um simulador interativo de missões espaciais em que o jogador/usuário deve planejar, executar e otimizar o envio de sondas e satélites a destinos no sistema solar, superando desafios de logística, comunicação, análise de dados, e otimização de rotas e recursos. O sistema é dividido em módulos que incorporam cada um dos conceitos exigidos.

SUMÁRIO

Introdução e Concepção.....	5
Tema.....	5
Justificativa.....	5
Visão geral das funcionalidades.....	5
Integração da imenta.....	6
Tecnologias escolhidas.....	6
Referências Bibliográficas.....	7
Anexo.....	8
A.1.....	8

Introdução e Concepção

Tema:







Simulador de Missões Espaciais Inteligentes - **AstroSim**

Justificativa:

O projeto AstroSim surge como uma ferramenta educacional interativa, projetada para simular missões espaciais com foco na aplicação prática de algoritmos e estruturas de dados. A escolha pelo tema “missões espaciais” se justifica por sua capacidade de incorporar de maneira natural e motivadora, os tópicos avançados do curso, como algoritmos gulosos, programação dinâmica, compressão de dados e grafos, tudo em um contexto lúdico e realista. O espaço representa um domínio, naturalmente, desafiador, ideal para demonstrar a complexidade computacional envolvida em decisões de rota, comunicação e análise de dados.

Visão Geral das Funcionalidades:

O software AstroSim terá os seguintes módulos principais:

-  Planejamento de Missão: Escolha de tarefas sob restrições (tempo, energia, carga), resolvendo problemas NP-completos como a Mochila com Restrições.
-  Inventário de Componentes: Sistema de busca por peças e dados usando algoritmos de busca (sequencial, binária, hashing, textual).
-  Transmissão de Dados: Módulo de compressão com algoritmos RLE e Huffman, simulando comunicação entre nave e Terra.
-  Mapa Espacial: Visualização e manipulação de rotas interplanetárias e redes de comunicação com grafos (Dijkstra, Ford-Fulkerson).
-  Gerenciamento de Recursos: Módulo de decisões gulosas para janelas de comunicação e alocação de combustível e carga.
-  Coleta e Análise de Dados Científicos: Módulo com algoritmos de programação dinâmica como Mochila 0/1, LCS, multiplicação de matrizes.





Integração da Ementa:

Cada tópico da ementa está vinculado a um módulo prático do sistema (Extrutura - Anexo I):

✓ Teoria da Complexidade (NP-Completo / NP-Difíceis):

- Módulo: Planejamento de Missão com Restrições
- Descrição: Escolha do melhor conjunto de tarefas (exploração, coleta, análise) respeitando limites de tempo, energia e carga.
- Problema usado: Problema da Mochila com Restrições Múltiplas — NP-completo.
- Aplicação: Exibição de aviso ao jogador de que o problema é intratável para grandes inputs e uso de heurísticas ou algoritmos aproximativos para encontrar boas soluções.
- Técnica extra: Permitir comparação entre solução ótima (via branch-and-bound) e heurística gulosa, para ilustrar complexidade.

✓ Algoritmos de Busca (Sequencial, Binária, Texto, Hashing):

- Módulo: *Banco de Dados Científicos / Inventário de Componentes*
- Aplicações:
 - Sequencial: Procura geral no inventário.
 - Binária: Busca por ID de componente em listas ordenadas.
 - Busca em Texto: Encontrar documentos relevantes em logs de missão (com KMP ou Boyer-Moore).
 - Hashing: Acesso rápido a sensores/códigos de componente via chave hash (tabela hash aberta/fechada).
- Interface: Campo de busca com tipo de algoritmo selecionável.

✓ Compressão de Dados (RLE, Huffman):

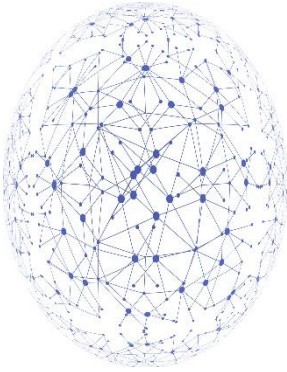
- Módulo: *Sistema de Transmissão de Dados da Nave para a Terra*
- Descrição: Transmitir dados de forma otimizada com baixo custo energético.
- Aplicações:
 - RLE: Compressão simples de imagens com áreas redundantes.
 - Huffman: Compressão de logs científicos ou mensagens.
- Extra: Comparação de taxa de compressão entre os algoritmos.





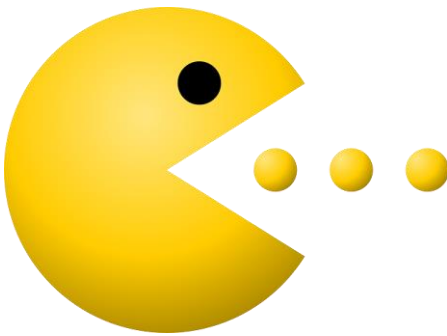
✓ Grafos (Caminhos, Fluxos, Conexões, entre outros):

- Módulo: *Navegação Interplanetária e Redes de Comunicação*
- Aplicações:
 - *Grafos de Caminhos Mínimos (Dijkstra/A):** Planejar rota com menor custo energético.
 - Fluxo Máximo (Ford-Fulkerson): Gerenciar capacidade de transmissão em redes de satélites.
 - Conectividade: Verificação de cobertura entre sondas (algoritmo de Tarjan).
 - Grafos Orientados: Dependência entre etapas da missão.
- Visual: Interface com mapa espacial onde o grafo é desenhado e interativo.



✓ Algoritmos Gulosos:

- Módulo: *Gerenciamento de Recursos da Nave*
- Problemas implementados:
 - Problema do Troco: Distribuição ideal de combustível restante.
 - Escalonamento de Intervalos: Escolha de janelas de comunicação com melhor retorno científico.
 - Mochila Fracionário: Otimizar carga científica levando apenas partes dos dados mais valiosos.
- Interface: Relatórios com comparação entre solução gulosa e outras estratégias.



✓ Programação Dinâmica

- Módulo: *Estratégia de Lançamento e Coleta de Dados*
- Problemas implementados:
 - Mochila 0/1: Escolha de instrumentos científicos respeitando peso e energia.
 - LCS (Longest Common Subsequence): Comparação de sequências de DNA alienígenas para pesquisa.
 - Multiplicação de Cadeias de Matrizes: Otimização de operações matemáticas na análise de dados.
- Interface: Painel com feedback visual mostrando subproblemas resolvidos.

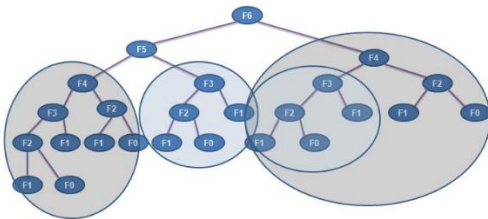


Tabela resumo:

Tópico da Ementa	Módulo no AstroSim
Teoria da Complexidade	Planejamento de missão com múltiplas restrições (mochila NP-completa)
Algoritmos de Busca	Inventário de componentes e logs com buscas variadas
Compressão de Dados	Comunicação entre nave e Terra via RLE e Huffman
Grafos	Mapeamento de rotas, comunicação e conexões entre sondas
Algoritmos Gulosos	Gerenciamento de combustível e tempo em decisões rápidas
Programação Dinâmica	Estratégia de coleta de dados científicos e análise de sequências

Tecnologias Escolhidas:

- i. Linguagem de Programação: Python 3.13+
- ii. Bibliotecas e Frameworks:

Área	Ferramentas
Interface gráfica	Tkinter + ttkbootstrap (ou PyQt6)
Grafos	NetworkX
Algoritmos	heapq, itertools, functools, collections
Compressão	zlib, base64
Visualização	matplotlib
Utilitários	numpy, scipy

- iii. Motivo da escolha:

Python é altamente expressivo, ideal para prototipagem rápida, possui vasto suporte algorítmico e ótima integração com bibliotecas de visualização e GUI. Além disso, é amplamente utilizado no meio acadêmico e científico, sendo uma escolha natural para um simulador educacional.



Referências Bibliográficas



1. CORMEN, Thomas H.; LEISERSON, Charles E.; RIVEST, Ronald L.; STEIN, Clifford. **Algoritmos: teoria e prática**. 3. ed. Rio de Janeiro: GEN LTC, 2013.
2. LAFORE, Robert. **Estruturas de dados e algoritmos em Java**. 2. ed. São Paulo: Pearson Education do Brasil, 2004.
3. GOODRICH, Michael T.; TAMASSIA, Roberto. **Estrutura de dados e algoritmos em Python**. Rio de Janeiro: LTC, 2017.
4. GAREY, Michael R.; JOHNSON, David S. **Computers and intractability: a guide to the theory of NP-completeness**. San Francisco: W.H. Freeman, 1979.
5. SIPSER, Michael. **Introdução à teoria da computação**. 3. ed. São Paulo: Cengage Learning, 2013.
6. SALOMON, David; MOTTA, Giovanni. **Data compression: the complete reference**. 4. ed. London: Springer, 2007.
7. SOMMERVILLE, Ian. **Engenharia de software**. 10. ed. São Paulo: Pearson, 2019.
8. PRESSMAN, Roger S. **Engenharia de software: uma abordagem profissional**. 8. ed. São Paulo: McGraw-Hill, 2016.
9. RUSSELL, Stuart; NORVIG, Peter. **Inteligência artificial**. 3. ed. São Paulo: Pearson, 2013.

ANEXO I

A.1

Estrutura Técnica

Estrutura Modular

AstroSim/

- |
- |— main.py (navegação geral)
- |— gui/ (interface com tkinter ou PyQt)
- |— modules/
 - | |— planner.py (complexidade)
 - | |— search_engine.py
 - | |— compression.py
 - | |— graph_manager.py
 - | |— greedy_solver.py
 - | |— dynamic_solver.py
- |— assets/
 - | |— planet_data/
 - | |— component_images/
- |— docs/ (ajuda e relatórios)

Diagrama de módulos

