

Assignment 1 Guide

Search & Problem Solving Agent (BFS, DFS, UCS) (Weeks 3–4)

Title: *Designing an Adaptive Puzzle Solver*

Objective Links: AI Fundamentals, Search and Problem Solving

Task:

- Define a custom search problem (not 8-puzzle or a standard maze). You should invent or adapt a domain (e.g., resource allocation on a small network, logistics scheduling, or a new board game).
- Implement two different search strategies in Python (one uninformed, one informed – e.g., UCS and A*) for solving a custom puzzle **that you define**.
- Evaluate and compare the algorithms on your domain: completeness, time/space complexity, and optimality.

Restrictions:

- Problem domain must be your own or a significant adaptation.
- Copying standard puzzle implementations will result in heavy penalty.

Deliverables:

- Source code in Python (via GitHub private repo).
- Short report (max 3 pages) summarizing algorithms, performance, and insights.

Note: Avoid just copy a previous implementation; you have to invent a problem domain, which forces understanding. Show your references that we can figure out what improvement you added.

Rubrics

Assignment 1 – Search & Problem Solving Agent (100 pts)

Criterion	Points	Description
Problem Domain Definition	15	Domain is non-trivial, clearly explained, and not a standard example (no 8-puzzle, classic mazes).
Correct Implementation of Algorithms	25	Two search strategies (one uninformed, one informed) implemented correctly.
Comparative Evaluation	20	Completeness, time/space complexity, and optimality measured and discussed.
Analysis & Justification	15	Clear explanation of why the domain/algorithms were chosen; insightful discussion of trade-offs.
Code Quality & Reproducibility	15	Readable, commented, runs without errors.
Report Quality (≤ 2 pages)	10	Well-structured, concise, and free of major language errors.

Plagiarism/Originality Rule: If the domain is obviously copied from a standard template, a maximum of 50% of the points can be awarded even if the code runs.