Assignment 1: Search Algorithms

Xavier F.C. Sánchez-Díaz Aleksandra Jekic

19th August 2025

Deadline: 12.09.2024, 23:59 hrs

Overview

In this assignment, you will experiment with the search algorithms we covered in the lecture. This is an exercise set where you will manually generate the list of expanded nodes and get a feel of the differences between various informed and uninformed search strategies.

The assignment consists of **eight tasks** divided into **three parts**. The **first part** is uninformed search, the **second part** will cover informed search, and the third part will focus on A^* search, admissibility and consistency. You need to make a **passable effort** on at least **two parts** to get this **assignment approved**. Each section specifies a set of required tasks you must do, either on your own or teaming up with at most one teammate.

1 The Problem: Speedrunning Routes

Are you a gamer? Some gamers push their abilities to the limit by completing the game as fast as possible. Here you will look for possible routes to complete a map from Doom, the classic shooter game from 1993, aiming to find the fastest route to the exit. Although modelling the map is an interesting way to practice our skills in data structures and mathematics, we have done that for you. We have conveniently provided a visualisation of this abstraction in Figure 1.

In Figure 1, every vertex represents an interest point, and every edge is an action (or movement). All actions have a cost of 1, but as you may notice, some of the edges have obstacles and thus the cost of traversing them will increase. Table 1 includes the costs for each obstacle. The starting state is A, marked with an arrow. The goal state is F, marked with a double ring. The heuristic value $h(n)$ for each state n is inside the floating circle, so $h(A) = 25$ and $h(I) = 22$.

Table 1: Costs of actions in the graph.

Obstacle	**Cost**
Stairs	1
Lift	2

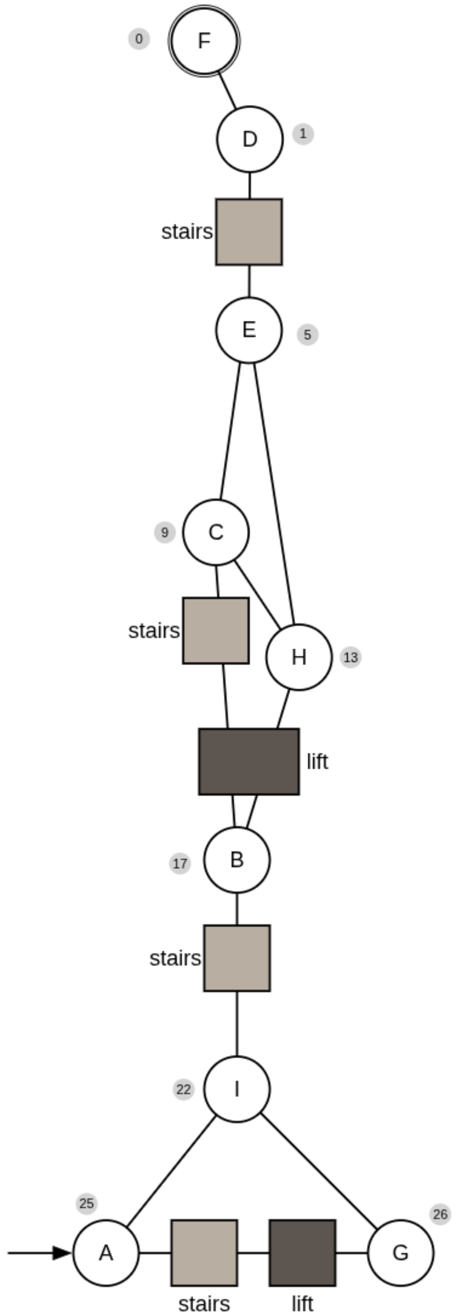

Figure 1: An abstract representation of E1M8: Phobos Anomaly. For simplicity, the states have been labelled from A to I so you can refer to them in your solutions.

2 Tasks

2.1 Part One: Setup and Uninformed Search

1. Set up the costs of actions in the graph. Consider the following costs as a starting point: $(A, G) = 4, (A, I) = 1, (B, H) = 3$

2. Execute the search on the graph shown in Figure 1 for the given algorithms below.

   a) Breadth-first search (BFS)
   b) Depth-first search (DFS)
   c) Uniform-cost search (UCS)

Write down the order of expansion, the path from start to goal (or write *None* if no path to goal is found), and write the cost of the path found. For example:

Example of output

BFS:

- Order of expansion: S, A, B, (G)
- Found path: SBG
- Path cost: 4

You can include your search trees as diagrams as well, but it is not necessary.

2.1.1 Assumptions

- Follow the pseudocode from the book:
  - For BFS, use `Breadth-First-Search` from **Figure 3.9 (p. 95)**
  - For DFS and UCS (and in fact, for all other algorithms in Parts Two and Three), use `Best-First-Search` from **Figure 3.7 (p. 91)** with their appropriate heuristic functions and data structures.
- Process (or generate) nodes in alphabetical order
  - So, C gets processed before H when B gets expanded.
- In case of a tie, use alphabetical order as a tie breaker
  - For example, if both C and D are equally good candidates for expansion, C should be expanded first.

2.2 Part Two: Informed Search

3. Execute the search on the graph shown in Figure 1 for the given algorithms below.

   a) Greedy Best-First Search (GBFS)

b) A^* Search

As with the previous part, write down the order of expansion, the path from start to goal (or write *None* if no path to goal is found), and write the *cost* of the path found.

2.3 Part Three: A^* Search, Admissibility and Consistency

While running A^* search you may have noticed that the result was not what you would have expected from an algorithm that is guaranteed to find the optimal solution. Why was this the case?

4. What mathematical property needed for optimality is not present in our graph? Explain intuitively why it is needed.
5. Prove (or disprove) this mathematical property.

Consider an updated heuristic function $h'(n)$, as shown in Figure 2.

6. Execute the search again, now on the updated and simplified graph (shown in Figure 2) using A^* search. Consider the same cost table you calculated in Part One, including the obstacles.
7. Is the heuristic $h'(n)$ admissible? Prove (or disprove) that it is admissible and discuss what it implies.
8. Is the heuristic $h'(n)$ consistent? Prove (or disprove) that it is consistent and discuss what it implies.

2.3.1 Bonus Task

9. How would you have to change the code of the A^* algorithm to make consistency necessary for cost optimality? What may you gain from making the algorithm less robust?

3 Deliverables

You must upload a **single** PDF containing **typeset** text, equations, and/or diagrams of your solutions.

- Include **natively digital** equations and diagrams. You can as well use hand writing on an tablet as long as it is readable. Do **not** upload scans or photos of hand-written solutions as these will be ignored by the TAs
- You can typeset your equations in Microsoft Word/Google Docs. If you're feeling adventurous, try LaTeX. Overleaf is a great place to start

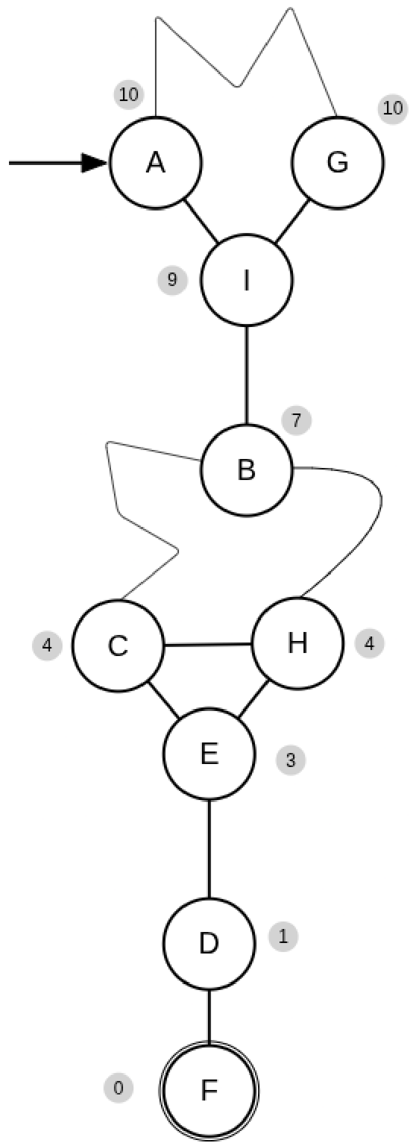

Figure 2: Updated and simplified representation of E1M8: Phobos Anomaly, with new heuristic values $h'(n)$. The edges have been adjusted visually to consider also the costs of the obstacles.