

Introduction to Artificial Intelligence (INFO8006)

Exercises 1 – Solving problems by searching

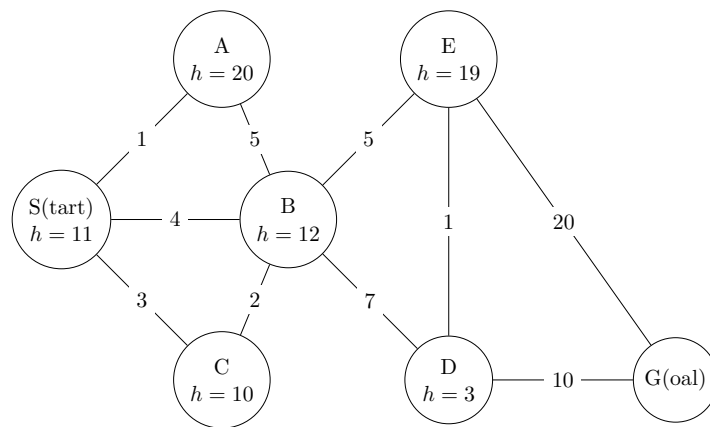
December 6, 2023

Learning outcomes

At the end of this session you should be able to

- formulate search problems rigorously;
- analyze the algorithms to perform uninformed search (depth-first, breadth-first, uniform-cost) and informed search (greedy, A*);
- apply these algorithms in fully observable and deterministic environments.

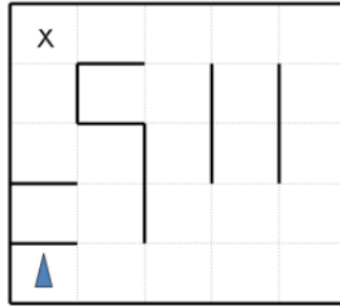
Exercise 1 Search algorithms



For each of the following search algorithms, apply the GRAPH-SEARCH algorithm (lecture 2, slide 62) and give the order in which the states, represented by the nodes of this undirected graph, are expanded, as well as the final path returned by the algorithm. If two nodes are in competition to be expanded, the conflict is resolved by alphabetical order.

1. Depth-First Search (DFS)
2. Breadth-First Search (BFS)
3. Uniform-Cost search (UCS)
4. Greedy search
5. A*
 - (a) Is the heuristic admissible ? If not, make it admissible.
 - (b) Is the heuristic consistent ? If not, can we apply GRAPH-SEARCH ?

Exercise 2 Maze Car (UC Berkeley CS188, Spring 2014)



Consider a car agent which has to exit a maze. At all time-steps, the agent points at direction $d \in \{N, S, E, W\}$. With a single action, the agent can either move forward at an adjustable velocity $v \in [0, V]$ or turn. The turning actions are **left** and **right**, which change the agent's direction by 90 degrees. Turning is only permitted when the velocity is zero (and leaves it at zero). The moving actions are **faster** and **slower**. The action **faster** increments the velocity by 1 and **slower** decrements the velocity by 1; in both cases the agent then moves a number of squares equal to its *new* velocity. Any action that would result in a collision with a wall is illegal. Any action that would reduce v below 0 or above a maximum speed V is also illegal. The agent's goal is to find a plan which parks it (stationary) on the exit square using as few actions (time steps) as possible.

As an example, in the hereabove maze, if the agent is initially stationary, it might first turn to the east using **right**, then move one square east using **faster**, then two more squares east using **faster** again. However, the agent will have to slow to take the turn.

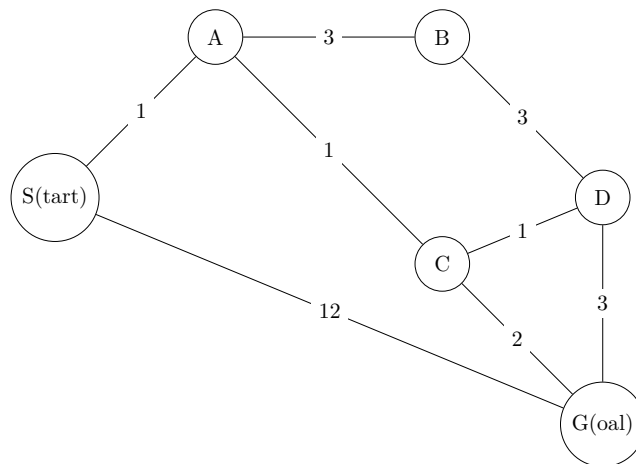
1. For a grid of size $M \times N$, what is the size of the state space? Assume that all configurations are reachable from the starting state.
2. What is the maximum branching factor of this problem? Assume that illegal actions are simply not returned by the successor function.
3. Is the Manhattan distance from the agent's location to the exit's location admissible?
4. If we used an inadmissible heuristic in A* tree search, could it change the completeness of the search? And the optimality?
5. State and motivate a non-trivial admissible heuristic for this problem.
6. Give a general advantage that an inadmissible heuristic might have over an admissible one.

Exercise 3 Heuristics (UC Berkeley CS188, Spring 2019)

Consider a graph search problem where all edges have a unit cost and the optimal solution has cost C^* . Let $h(n)$ be a heuristic which is $\max\{h^*(n) - k, 0\}$, where $h^*(n)$ is the true forward cost of n and $k \leq C^*$ is a non-negative constant.

1. Is h admissible?
2. Which of the following is the most reasonable description of how much more work will be done, *i.e.* how many more nodes will be expanded, with heuristic h compared to h^* , as a function of k ?
 - (a) Constant in k
 - (b) Linear in k
 - (c) Exponential in k
 - (d) Unbounded

Exercise 4 Search algorithms



For each of the following search algorithms, give the order in which states are expanded as well as the final path returned by the algorithm. If two nodes are in competition to be expanded, the conflict is resolved by alphabetical order.

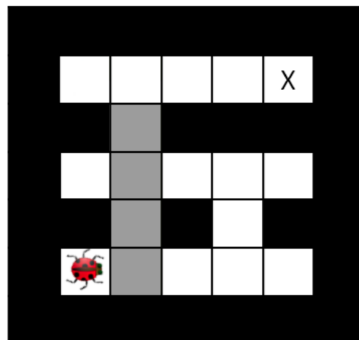
1. Depth-First search (DFS)
2. Breadth-First Search (BFS)
3. Uniform-Cost search (UCS)
4. Consider the following heuristics: which one is not admissible? Why?

State	h_1	h_2
S	5	4
A	3	2
B	6	6
C	2	1
D	3	3
G	0	0

5. A* (with the admissible heuristic)

Exercise 5 The hive (UC Berkeley CS188, Spring 2019)

The hive of insects needs your help. You control an insect in a rectangular maze-like environment of size $M \times N$, as shown on the Figure below. At each time-step, the insect can move into a free adjacent cell or stay in its current location. All actions have a unit cost.



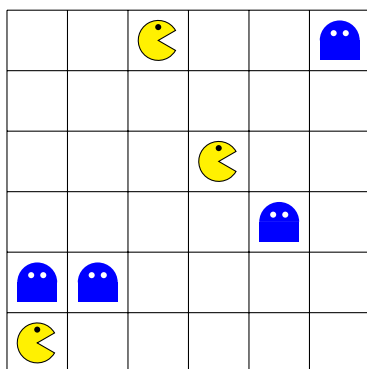
In this particular case, the insect must pass through a series of partially flooded tunnels, as illustrated by the gray cells on the map. The insect can hold its breath for A time-steps in a row. Moving into a flooded cell requires your insect to consume 1 unit of air, while moving into a free cell refills its air supply.

- Give a minimal state space for this problem (do not include extra information). You should answer for a general instance of the problem, not the specific map above.
- Give the size of your state space.

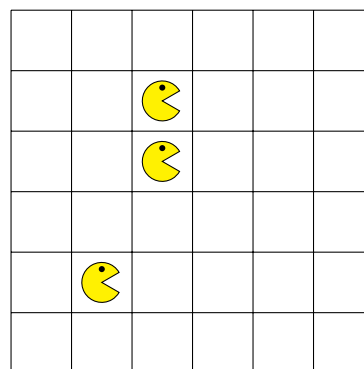
Exercise 6 Pacmen (January 2023)

Pacman and his friends have decided to combine forces and go on the offensive, and are now chasing ghosts! In a grid of size $M \times N$, Pacman and $P - 1$ of his friends are moving around to collectively eliminate all of the ghosts in the grid by stepping on the same square as each of them. Moving into the same square as a ghost will eliminate it from the grid. At every turn, Pacman and his friends may choose one of the four (**north**, **south**, **east**, **west**) actions but may not collide with each other. In other words, any action that would result in two or more Pacmen occupying the same square will result in no movement for either of the Pacmen. Additionally, Pacman and his friends are *indistinguishable* from each other. There are a total of G ghosts, which are also indistinguishable from each other and cannot move.

Treating this as a search problem, we consider each configuration of the grid to be a state, and the goal state to be the configuration where all ghosts have been eliminated from the board. The cost of each turn (all Pacmen move once) is 1. Below is an example starting state, as well as an example of goal state.



Initial state example



Final state example

Suppose that Pacman has no friends ($P = 1$).

1. What is the size of the minimal state space representation? Explain your answer.
2. Explain whether each of the following heuristics is admissible, consistent, neither, or both.
 - $h_1(s)$ = the sum of the Manhattan distances from Pacman to every ghost.
 - $h_2(s)$ = the number of ghosts times the maximum Manhattan distance between Pacman and any of the ghosts.
 - $h_3(s)$ = the number of remaining ghosts.

Suppose that Pacman has exactly one less friend than the number of ghosts ($P = G$).

3. What is the size of the minimal state space representation? Explain your answer.
4. Explain whether each of the following heuristics is admissible, consistent, neither, or both.
 - $h_4(s)$ = the largest of the Manhattan distances between each Pacman and its closest ghost.
 - $h_5(s)$ = the smallest of the Manhattan distances between each Pacman and its closest ghost.
 - $h_6(s)$ = the number of remaining ghosts.
 - $h_7(s)$ = the number of remaining ghosts divided by P .

Supplementary materials

- Search (UC Berkeley CS188, Spring 2014 Section 0).



- Search (UC Berkeley CS188, Spring 2014 Section 1).



- Chapter 3 of the reference textbook.