INFO8006 - Introduction to Artificial Intelligence

Exercise session 1

Search problem formulation

A search problem is defined by

- A representation for states.
- The initial state of the agent.
- A set of actions allowed in every state s.
- A transition model $s' = result(s, a)$ that returns the resulting state s' for using action a in state s.
- A goal test which determines if the problem is solved in state s.
- A path cost that assigns a numerical cost to each path. We assume the later to be a sum of positive step costs $c(s, a, s')$ along the path.

Searching strategy

- **Depth-First Search** uses a stack (LIFO) for the *fringe*.
- **Breadth-First Search** uses a queue (FIFO) for the *fringe*.
- **Uniform-Cost Search** uses a priority queue for the *fringe*, where the priority is the cumulative cost $g(n)$.
- **Greedy Search** uses a priority queue for the *fringe*, where the priority is a heuristic $h(n)$.
- **A* Search** uses a priority queue for the *fringe*, where the priority is the sum of both heuristic $h(n)$ and cumulative cost $g(n)$.

Heuristic

A heuristic function $h(n)$ estimates the cost of the cheapest path from node n to a goal state.
A heuristic is admissible if

$$0 \leq h(n) \leq h^*(n)$$

with $h^*(n)$ being the tightest heuristic, i.e. the true cumulative cost.
A heuristic is consistent if

$$h(n) \leq c(n, a, n') + h(n')$$

for all node-action-resulting node triplets (n, a, n').

```
function GRAPH-SEARCH( problem, fringe) returns a solution, or failure
    closed ← an empty set
    fringe ← INSERT(MAKE-NODE(INITIAL-STATE[problem]), fringe)
    loop do
        if fringe is empty then return failure
        node ← REMOVE-FRONT(fringe)
        if GOAL-TEST(problem, STATE[node]) then return node
        if STATE[node] is not in closed then
            add STATE[node] to closed
            fringe ← INSERTALL(EXPAND(node, problem), fringe)
    end
```

In session exercises: Ex. 1, Ex. 2.1-3, Ex. 3

Exercise 1 Path finding

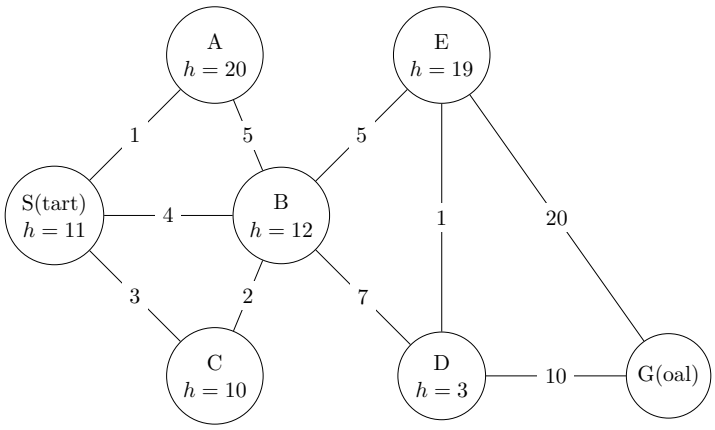
Tram'Ardent deployed a new robot to help them on the finishing touches of Liège worksite. This robot's goal is to convey weeds that were removed by the gardeners to a container as fast as possible. Assuming that

- the fuel cost used by the robot is negligible
- the robot can only navigate on guides that are placed in a rectangular grid configuration and can locate itself on the grid
- there is only one pile and one container by zone (i.e. by grid) for which positions are known by the robot
- the speed of the robot is constant and the grass loading/unloading is instantaneous
- the robot can only carry a limited weight of grass at once
- the robot can know the initial weight of grass on the pile.

For a given grid

1. Formulate a corresponding search problem.
2. Knowing that the terrain can be rugged at specific places, causing slow down of the robot, propose an admissible heuristic which estimates the remaining cost at any node.
3. Now the robot can move in diagonal. Is your heuristic still admissible? If not, what should you change? What if the robot is twice as fast diagonally?

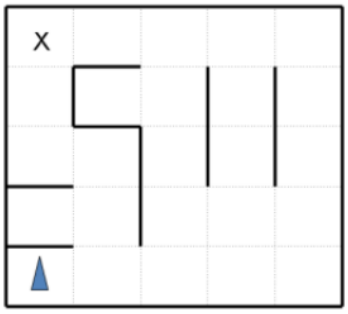
Exercise 2 Search algorithms



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 assigning priority according to 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 3 Maze Car (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.

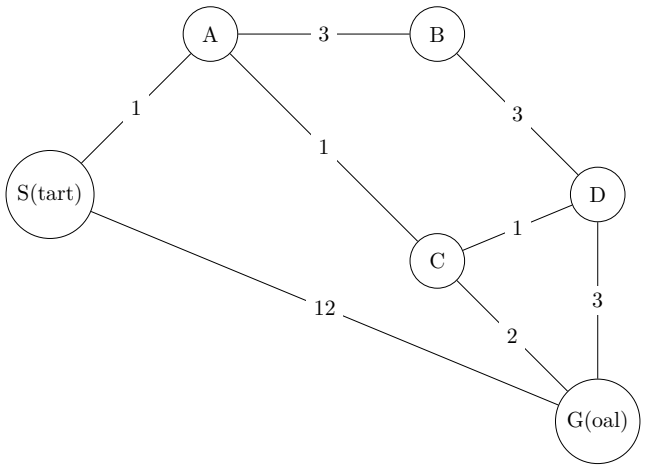
1. For a grid of size $M \times N$, what is the size of the state space? Assume that all configurations are reachable from the starting state.
2. What is the maximum branching factor of this problem? Assume that illegal actions are simply not returned by the successor function.
3. Is the Manhattan distance 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 4 Heuristics (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 5 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.

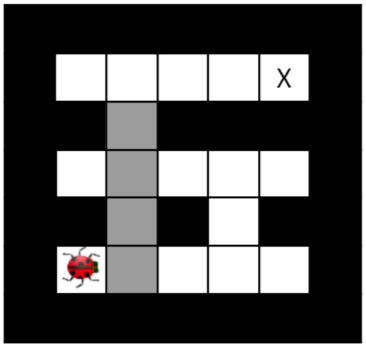
1. Depth-First search (DFS)
2. Breadth-First Search (BFS)
3. Uniform-Cost search (UCS)
4. Consider the following heuristics: which one is not admissible? Why?

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 6 The hive (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.



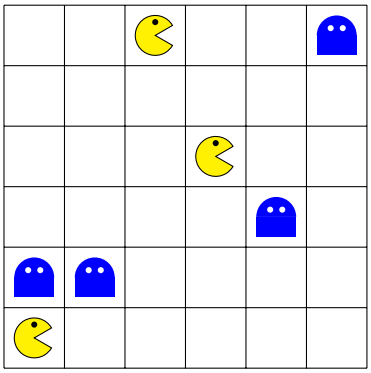
In this particular case, the insect must pass through a series of partially flooded tunnels, as illustrated by the gray cells on the map. The insect can hold its breath for A time-steps in a row. Moving into a flooded cell requires your insect to consume 1 unit of air, while moving into a free cell refills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

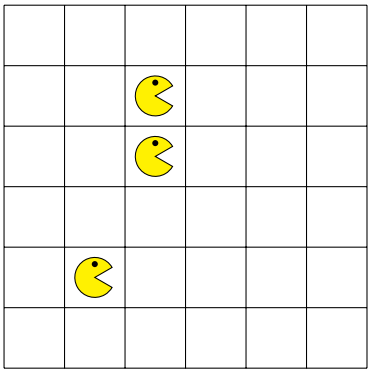
Exercise 7 Pacmen (INFO8006, January 2023)

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.

Quiz

The A*algorithm, in its graph version is ...

- ☐ always optimal.
- ☐ always optimal if the heuristic is admissible.
- ☐ always optimal if the heuristic is consistent.
- ☐ never optimal when there are cycles.

In a fully observable but stochastic game, if a simple reflex agent is presented the same state several times, it will ...

- ☐ take the same action every time.
- ☐ take a different action every time.
- ☐ take a different action most of the time.
- ☐ take the same action most of the time.

Which of the following statements about heuristics are true ?

- ☐ $h(n) = 0 \ \forall n$ can speed up A*compared to UCS.
- ☐ If a heuristic is admissible, it is guaranteed to be consistent.
- ☐ The average step cost $h(n) = \frac{1}{\#a} \sum_a c(n, a, n')$ is admissible.
- ☐ The true remaining cost dominates all admissible heuristics.

The poker game is ...

- ☐ discrete, multi-agent and deterministic.
- ☐ partially-observable, stochastic and multi-agent.
- ☐ partially-observable, episodic and single-agent.
- ☐ None of the above.

A goal-based planning agent ...

- ☐ cannot decide which action to take next if the transition model is wrong.
- ☐ ignores the consequences of its actions to take decisions.
- ☐ can be implemented using the greedy search algorithm.
- ☐ need a Markovian system to be optimal.