1 Solving problems by searching (26/09/2019)

1.1 Objectives

At the end of this exercise session you should be able to:

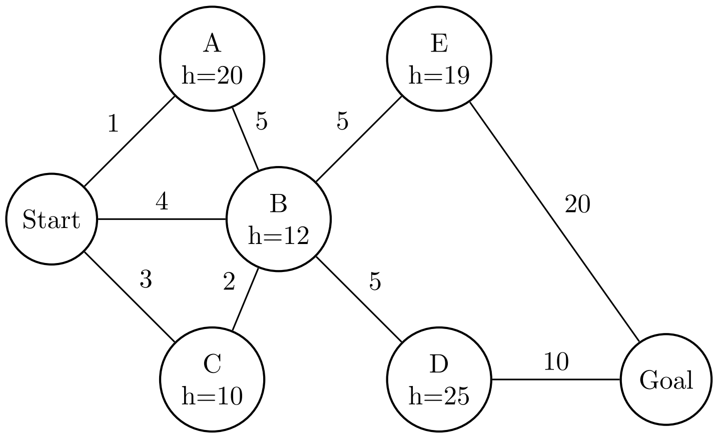
- Theoretically (COST[1]) analyse the algorithms to do uninformed (Depth-first, Breadth-first, Uniform-cost) and informed (Greedy-search, A-star) search.
- Be able to apply each of these algorithms on any search problem define in a fully observable and deterministic environment.

1.2 Theory

See slides.

1.3 Exercises

1.3.1 Search algorithms (≈ 60 min)



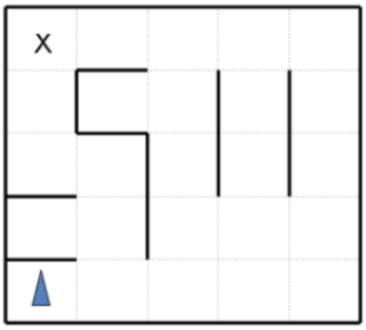
For each of the following search algorithms, give the order in which states are expanded as well as the final path returned by the algorithm. If two nodes are in competition to be expanded, the conflict is resolved by alphabetical order.

1. Depth-First
2. Breadth-First
3. Uniform-Cost
4. Greedy
5. A-Star (Is the heuristic admissible?)

1.3.2 Maze Car (≈ 25 min)

This exercice is taken from `http://ai.berkeley.edu/sections/section_1_0hzy6TFupb1Z3bckfRXdC5KYpsdZOE.pdf`.
Imagine a car-like agent wishes to exit a maze like the one shown below:

[1] Completeness-Optimality-Space Complexity-Time Complexity



The agent is directional and at all times faces some direction d ∈ (N, S, E, W). With a single action, the agent can either move forward at an adjustable velocity 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 fast and slow. Fast increments the velocity by 1 and slow decrements the velocity by 1; in both cases the agent then moves a number of squares equal to its NEW adjusted velocity. Any action that would result in a collision with a wall crashes the agent and is illegal. Any action that would reduce v below 0 or above a maximum speed V_{max} 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: if the agent shown were initially stationary, it might first turn to the east using (right), then move one square east using fast, then two more squares east using fast again. The agent will of course have to slow to turn.

- Quizz (≈ 15 minutes):
  1. If the grid is M by N, what is the size of the state space? You should assume that all configurations are reachable from the start state.
  2. What is the maximum branching factor of this problem? You may 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?
  5. If we used an inadmissible heuristic in A* tree search, could it change the optimality of the search?
- Discussion (if time permits) (≈ 10 minutes):
  1. State and justify a non-trivial admissible heuristic for this problem.
  2. Give a general advantage that an inadmissible heuristic might have over an admissible one.

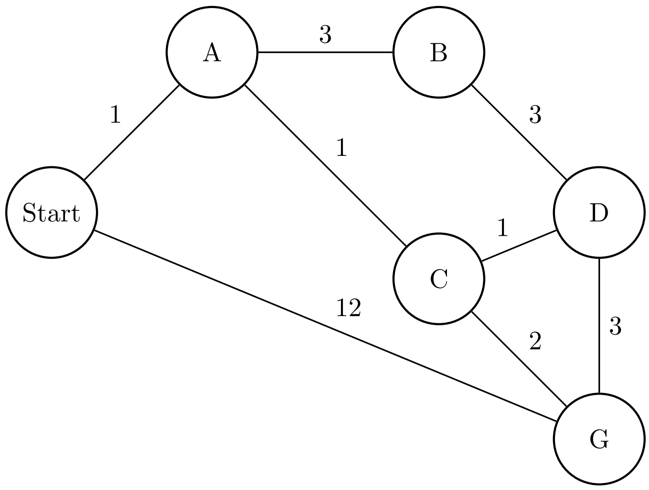
1.4 Supplementary exercises

1.4.1 Heuristic

Consider a search problem where all edges have cost 1 and the optimal solution has cost C. Let h be a heuristic which is $\max\{h^* - k, 0\}$, where h^* is the actual cost to the closest goal and k is a nonnegative constant.

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

1.4.2 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
2. Breadth-First
3. Uniform-Cost
4. Consider the following heuristics: Which one is not admissible? Why?

State	h1	h2
S	5	4
A	3	2
B	6	6
C	2	1
D	3	3
G	0	0

Table 1: 2 possible heuristics

5. Use the admissible algorithm to apply A* algorithm

1.4.3 The hive

The hive of insects needs your help. You control an insect in a rectangular maze-like environment with dimensions M x N, as shown on 1. At each time step, the insect can move into a free adjacent square or stay in its current location. All actions have cost 1.
In this particular case, the insect must pass through a series of partially flooded tunnels. Flooded squares are lightly shaded in the example map shown. The insect can hold its breath for A time steps in a row. Moving into a flooded square requires your insect to expend 1 unit of air, while moving into a free square refills its air supply.

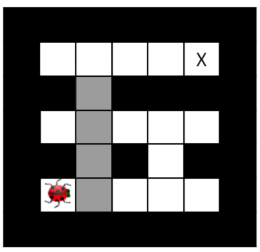


Figure 1:

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

1.5 Supplementary material

http://ai.berkeley.edu/sections/section_0_v55LOfoUUwiW1k6Nchnk3Dw6WQuTW8.pdf
http://ai.berkeley.edu/sections/section_1_0hzy6TFupb1Z3bckfRXdC5KYpsdZOE.pdf