Einführung in Artificial Intelligence SS 2025, 4.0 VU, 192.027

Exercise Sheet 2 – Problem Solving and Search

For the discussion part of this exercise, mark and upload your solved exercises in **TUWEL** until Wednesday, June 11, 23:55 CEST. The registration for a solution discussion ends on Friday, June 13, 23:55 CEST. Be sure that you tick only those exercises that you can solve and explain!

In the discussion, students will be asked questions about their solutions of examples they checked. The discussion will be evaluated with 0–25 points, which are weighted with the fraction of checked examples and rounded to the next integer. There is *no minimum number of points* needed for a positive grade (i.e., you do not need to participate for a positive grade, but you can get at most ≈80% without doing exercises).
Note, however, that *your registration is binding*. Thus, *if* you register for a solution discussion, then it is *mandatory* to show up. No-show after a registration without plausible excuse leads to a penalty. Such students will not have the possibility to participate in another exam once they have gotten a certificate. If you registered but cannot show up due to unpredictable and unavoidable obstacles, either deregister or send us a confirmation (doctor's note, etc) and you will be excused. Please ask questions in the **TUWEL** forum or visit our tutors during the tutor hours (see **TUWEL**).

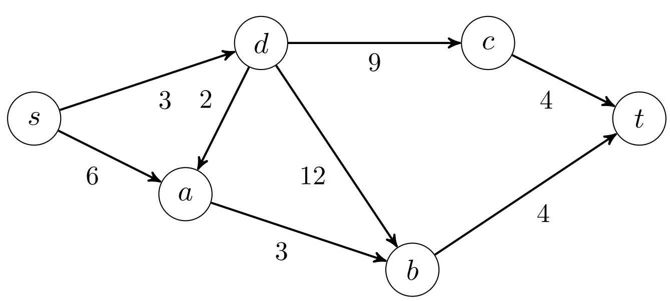
Exercise 2.1: Consider again the 8-Puzzle discussed in the lecture. Consider the discussed heuristics

(a) $h_1(n)$: number of misplaced tiles,

(b) $h_2(n)$: Manhattan distance.

Show whether the two suggested heuristics are admissible and/or consistent (monotonic).

Exercise 2.2: Perform the greedy algorithm and the A* algorithm (tree search version) using the given heuristic function h on the following graph in order to find a shortest path from s to t. In which order are the nodes expanded? Show the contents of the priority queue at each iteration. If multiple nodes have the same priority, expand the one that comes first alphabetically. Compare the shortest paths returned by the two algorithms, argue whether these solutions are optimal? Justify your answer.



$$h(s) = 12, h(a) = 6, h(b) = 4, h(c) = 3, h(d) = 7, h(t) = 0$$

Exercise 2.3: Assume h_a, h_b are admissible heuristics. Which of the following heuristics is admissible? Justify your answer.

(a) $h_1(n) = h_a(n) \cdot h_b(n)$.

(b) $h_2(n) = \frac{h_a(n)+h_b(n)}{2}$

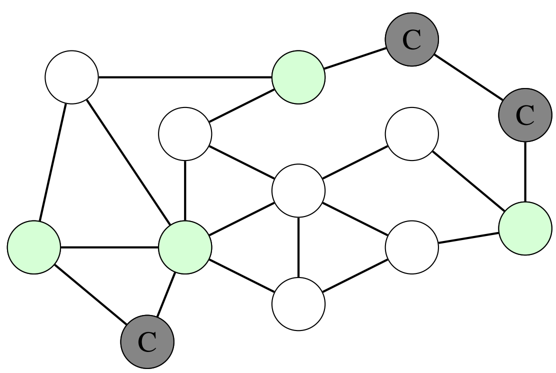
Exercise 2.4: Provide a proof or a counterexample for each of the following claims:

(a) Every consistent heuristic (with $h(G) = 0$ for each goal node) is admissible.

(b) Every admissible heuristic is consistent.

Please try to keep your counterexample(s) as simple as possible. The graph(s) you use should have no more than five nodes.

Exercise 2.5: Newly paranoid after the defeat on Weathertop, Sauron has constructed additional watchtowers all over Mordor and will not allow anyone but the Nazgûl to be stationed there. However, only three of The Nine can be spared to man the towers, as the rest are actively pursuing the Ring-bearer. The area controlled by a given watchtower is considered secure if a Nazgûl is stationed there or at a neighbouring tower. The roads connecting the watchtowers are depicted as edges in the following graph, with the towers as nodes. Sauron randomly assigned the Nazgûl to the towers, so now it's your turn to execute a hill-climbing local search to find a goal state where each tower is covered. A step in the search process involves transferring one Nazgûl from one tower to another, the number of secured towers represents the value of each state. *Note: A Nazgûl can be transferred to any tower during a step, however only one can be transferred at a time.* Illustrate your answer by drawing graphs like the one below.



Covered Towers: 7

Exercise 2.6: Provide a proof or a counterexample for each of the following claims:

- Simulated annealing always outperforms exhaustive search.
- Crossover always helps in genetic algorithms.

Exercise 2.7: Consider the vacuum-cleaner world example discussed in the lecture. This time, a software update created a mess. The agent may think that square A cannot be dirtier than square B. It affects actions "Suck" and "Left" in the following way.

(1) When it is at square B, and square A is cleaned, it may not execute the "Suck" action. Instead, it moves back to square A, and makes square A dirty again along the way. Namely, the "Suck" action on state [B, , Dirt] results in either [B, ,] or [A, Dirt, Dirt].

(2) When it is at square B, where square B is cleaned and square A is dirty, do the "Left" action may also let it start cleaning automatically. Namely, the "Left" action on state [B, Dirt,] results in either [A, Dirt,] or [A, ,].

Decide whether this search problem with nondeterministic action effects can be solved with an *And-Or Tree* or *Cyclic Solution*. Illustrate the corresponding solution visually.

Exercise 2.8: Perform the Online Search algorithm, as outlined in the lecture, on the following maze. Show the agent's location in the maze for each state along with the next move and the updates made to the *untried*, *result*, and *unbacktracked* arrays. Note that the preferred moves are in the order of Up (U), Right (R), Left (L), and Down (D).