By February 4th, 2015 solutions for the following exercises have to be submitted: 1, 2, 3, 5, 7.

Exercise 1 : Heuristics
Let \( n \) be a node in a search space graph that is explored by \( A^* \) using heuristic \( h \).

(a) When can the value of \( h(n) \) change during \( A^* \)-search?
(b) When can the value of \( g(n) \) change during \( A^* \)-search?
(c) When can the value of \( g(n) \) change during \( A^* \)-search with admissible \( h \)?
(d) When can the value of \( g(n) \) change during \( A^* \)-search with monotone \( h \)?

Exercise 2 : Admissibility
In the lecture, we have proven that the algorithm \( A^* \) is admissible when using an admissible heuristic. Does this also apply to the algorithm \( Z \)?
Justify your answer.

Exercise 3 : Monotonicity
Consider the search problem of finding the shortest sequence of knight moves between two given squares on a chessboard, as discussed in the lecture. Let \( h \) be a heuristic that estimates the number of moves remaining for a given board square represented by node \( n \).

\[
h(n) = \frac{d_M(n, \gamma)}{3}
\]
where \( d_M(n_1, n_2) \) is the Manhattan distance between the board squares corresponding to nodes \( n_1 \) and \( n_2 \).

(a) Define the term “monotonicity” in the context of heuristic cost estimation functions.
(b) Show that \( h \) is both admissible and monotone.
(c) For the same search problem, find a heuristic that is admissible, but not monotone.

Exercise 4 : Reopening
Consider the following search space graph:

Add edge cost values and \( h \)-values for the nodes, such that the number of reopenings done by algorithm \( A^* \) is exponential in \( k \).
Exercise 5 : Relaxed Models

Consider the general approach to weighted node evaluation, where \( f_w(n) = (1 - w) \cdot g(n) + w \cdot h(n) \). Under which conditions does it make sense to use a weighting where \( w \) is close to 1, disregarding the current path cost \( g \)?

Exercise 6 : Relaxed Models

The algorithm \( A^*_\varepsilon \) uses two heuristic functions \( h \) and \( h_F \). What is the advantage of using \( h \neq h_F \)?

Exercise 7 : Implementing DWA* Search

Consider the 8-puzzle problem. Your task is to reach the goal state \( \gamma \) with as few moves as possible.

\[
\begin{array}{ccc}
6 & 4 & 7 \\
8 & 5 & \\
3 & 2 & 1 \\
\end{array}
\quad
\begin{array}{ccc}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & \\
\end{array}
\]

(s) Consider the case that \( \gamma \) would not be reachable from a start state. Would DWA* expand the same number of nodes as A* until it fails?

(b) Consider the case \( N = 1, \varepsilon \approx 1 \) and \( h \approx h^* \). Would you expect that DWA* performs more, less, or the same amount of node expansions as A*?

(c) Consider the case \( N \to \infty, \varepsilon \approx 1 \) and \( h \approx h^* \). Would you expect that DWA* performs more, less, or the same amount of node expansions as A*?

(d) Implement a DWA* search for the 8-Puzzle problem using \( h_2 \), the sum of the Manhattan distances from each of the 8 tiles to its position in the goal state. What changes do you need to make to your A* implementation? You can also use the A* example implementation from the lecture’s web page as a base.

(e) Run your implementation with \( N \in \{5, 10, 15, \ldots, 50\} \) and \( \varepsilon \in \{\frac{1}{8}, \frac{1}{4}, \frac{1}{2}, 1, 2\} \). Use s as the start and \( \gamma \) as the goal state. Show the number of node expansions performed by DWA* for each of the 50 pairs of \( N \) and \( \varepsilon \). You can use a table or graphs.

(f) For the same parameters as above, show the number of moves of the solution found by DWA*, \( g(\gamma) \), for each pair of \( N \) and \( \varepsilon \).

(g) For what parameter setting does DWA* terminate with an optimal solution with the smallest amount of node expansions? How many node expansions are these?

(h) For what parameter setting does DWA* terminate with a solution with the smallest amount of node expansions? How many node expansions are these?