DQ4 (L4)

Due No due date **Points** 22 **Questions** 7 **Time limit** None
Allowed attempts Unlimited

Instructions

- This quiz is NOT GRADED. However, it is HIGHLY RECOMMENDED that you use these questions to complement your review of the lecture content.
- The questions are based on content from the Lecture 4 and from part of Chapter 3 of the AIMA (4th Ed.) textbook (i.e., 3.6).

Take the quiz again

Attempt history

	Attempt	Time	Score
KEPT	Attempt 2	less than 1 minute	22 out of 22
LATEST	Attempt 2	less than 1 minute	22 out of 22
	Attempt 1	8 minutes	17 out of 22

Submitted 30 Jan at 13:28

Question 1

3 / 3 pts

Determine if the following statement is true or false.

Assume that a rook can move on a chessboard any number of squares in a straight line, vertically or horizontally, but cannot jump over other pieces. Manhattan distance is an admissible heuristic for the problem of moving the rook from square A to square B in the smallest number of moves.

True

Correct!

False

False. A rook can cross the board in one move, so Manhattan is pessimistic and hence not an admissible heuristic.

Specifically, in the following situation:

RXXXT

Where X denote blank square, R the position of the Rook and T the target, Manhattan distance would return 4, However, the Rook is actually only 1 move away.

Question 2

3 / 3 pts

For each of the following admissible heuristics of the 8-puzzle, match the **most accurate description** for it. Refer to the Lecture slides for a description of the 8-puzzle.

- $h_1(n)$ = A = Euclidean Distance
- $h_2(n)$ = B = Manhattan Distance
- $h_3(n)$ = C = Misplaced Tiles

Correct! **A = Euclidean Distance** A numbered tile may n

Correct! **B = Manhattan Distance** A numbered tile may n

Correct! **C = Misplaced Tiles** A numbered tile that is

As defined by the solution.

Question 3

6 / 6 pts

Given the heuristics defined in Question 2 (i.e., h_1, h_2, h_3), we further denote the following heuristics.

- $h_4(n) = h_1(n) + h_3(n)$
- $h_5(n) = \max(h_1(n), h_2(n))$
- $h_6(n) = \min(h_2(n), h_4(n))$
- $h_7(n) = \max(h_3(n), h_4(n))$

Select ALL the options that are true.

- [ ] h_4 is admissible.

Correct!

- [x] h_5 is admissible.

Correct!

- [x] h_6 is admissible.
- [ ] h_7 is admissible.

Given that h_1, h_2, h_3 are all admissible, we thus have the following.

Option 1: $h_4(n) = h_1(n) + h_3(n)$: ***h_4 is admissible*** --- ***This is not true.*** As a counter example, consider the case when n corresponds to a state while only one tile needs make one move in order to complete the 8-puzzle (i.e., $h^*(n) = 1$). In this case, $h_1(n) = 1$ and $h_3(n) = 1$, which results in $h_4(n) = 2$. However, since we only needed one move to complete the puzzle, h_4 has overestimated the number of moves required (i.e., $h_4(n) > h^*(n)$). As such, h_4 is NOT admissible.

Option 2: $h_5(n) = \max(h_1(n), h_2(n))$: ***h_5 is admissible*** --- ***This is true.*** As h_1 and h_2 are both admissible, for all n, $h_1(n) \leq h^*(n)$ and $h_2(n) \leq h^*(n)$. Therefore, for all n, $\max(h_1(n), h_2(n)) \leq h^*(n)$.

Option 3: $h_6(n) = \min(h_2(n), h_4(n))$: ***h_6 is admissible*** --- ***This is true.*** Even though h_4 is inadmissible (i.e., at least for some n, $h_4(n) > h^*(n)$), because h_2 is admissible (i.e., for all n, $h_2(n) \leq h^*(n)$, for all n, $\min(h_2(n), h_4(n))$ will be admissible since for those n where $h_4(n) > h^*(n)$, we have $h_2(n) \leq h^*(n) < h_4(n)$, and as such $h_6(n) = h_2(n)$.

Option 4: $h_7(n) = \max(h_3(n), h_4(n))$: ***h_7 is admissible*** --- ***This is not true***. We know that at least for one n, $h_4(n) > h^*(n)$; let us denote this case as n'. Therefore, even if for all n, $h_3(n) \leq h^*(n)$, we know that at n', we have $h_3(n) \leq h^*(n) < h_4(n)$, and as such we would have $h_7(n') = h_4(n')$, and consequently, $h_7(n') > h^*(n')$. Thus, h_7 is NOT admissible.

Question 4

3 / 3 pts

Consider the travelling salesman problem (TSP). In order to find a tour, it must have the following rules:

1. A graph with subset of edges
2. Connected
3. Total Length of Edges minimised
4. Each node has degree 2

Determining a minimum spanning tree (MST) is a relaxed problem of TSP. Which rule do we remove to achieve a MST?

- Rule 1
- Rule 2
- Rule 3
- Correct! Rule 4

As defined by the solution.

Question 5

2 / 2 pts

If a heuristic h_1 dominates another heuristic h_2, then for all states n, we have $h_2(n) \leq h_1(n) \leq h^*(n)$.

Consider if the above statement is true or false.

As stated in the lecture, assume that dominance implies admissibility (of both heuristics).

- Correct! True
- False

Some may argue that dominance may be considered without the assumption of admissibility — i.e., we may consider the dominance of non-admissible heuristics as well.

While this is certainly something that may be adopted, it is not the typical practice. Largely, dominance is used to distinguish between more and less efficient heuristics under the context of the A* informed search algorithm, which is why, generally, admissibility is assumed.

Question 6

3 / 3 pts

Given that a particular search solution expanded ***N*** nodes and found a solution at depth ***d***, then the effective branching factor, ***b****, for that search may be determined by solving for which of the following formulas?

Note, assume that the ***initial node*** is given, and was ***not derived via expansion***. - i.e., the initial node should not be considered as part of ***N***.

- $N((b^*) - 1) = (b^*)^d - 1$
- $N((b^*) - 1) = (b^*)^{d+1} - 1$
- Correct! $(N + 1)((b^*) - 1) = (b^*)^{d+1} - 1$
- None of the above

Assuming a constant branching factor b*, the complete tree at depth d would contain (b*)0 + (b*)1 + (b*)2 + ... + (b*)d-1 + (b*)d vertices.

We know that the sum of a geometric series is given by:
$a + ar + ar^2 + ... + ar^{(n-1)} = a.(r^n - 1) / (r - 1)$

Thus, given N nodes in the tree, and the solution is at depth d, we have must solve the following for b* to get the effective branching factor.
N + 1 = (b*)0 + (b*)1 + (b*)2 + ... + (b*)d-1 + (b*)d
$N + 1 = ((b^*)^{d+1} - 1) / ((b^*) - 1)$
$(N + 1).((b^*) - 1) = (b^*)^{d+1} - 1$

This is stated as Option 4.

Note that we have N + 1 on the RHS since we assume that the initial node is not part of N.

Question 7

2 / 2 pts

When the A* search algorithm utilises a heuristic h_1 that dominates another heuristic, h_2, the effective branching factor for the search conducted by A* using h_1 will never be greater than the effective branching factor for the search conducted by A* using h_2.

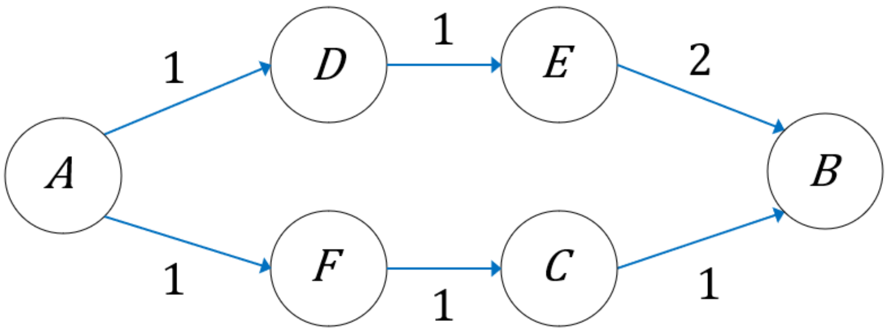
Determine if the above is true or false.

True

Correct!

False

This is **False**. Consider the following counterexample.



n	$h_1(n)$	$h_2(n)$	$h^*(n)$
A (init)	3	3	3
D	2	2	3
F	2	0	2
E	2	2	2
C	1	0	1
B (goal)	0	0	0

Assume that tie-breaking is resolved via ascending alphabetical ordering.

Trace for A* using h_1:
I1: [A(-), 0+3]
I2: [D(A), 1+2], [F(A), 1+2] // tie-break D over F
I3: [F(A), 1+2], [E(A,D), 2+2]
I4: [C(A,F), 2+1], [E(A,D), 2+2]
I5: [B(A,F,C), 3+0], [E(A,D), 2+2] // tie-break B over E
I6: Done (A,F,C,B)

Trace for A* using h_2:
I1: [A(-), 0+3]
I2: [F(A), 1+0], [D(A), 1+2]
I3: [C(A,F), 2+0], [D(A), 1+2]
I4: [B(A,F,C), 3+0], [D(A), 1+2] // tie-break B over D
I5: Done (A,F,C,B)

Notice that under h_1, one additional node (E), is generated, while the solution depth remains the same for both h_1 and h_2, which means that h_1 has resulted in a higher effective branching factor. This counterexample thus disproves the given statement.

Without tie-breaking, the statement holds. This was briefly described in Lecture 4. Can you prove this? Do attempt to do so as an added exercise when time permits.

Credit to ***Richard Willie*** for finding this counerexample.