# Week 2 - Blind Search and Systematic Heuristic Search

**Due** No due date **Points** 9 **Questions** 9 **Available** after Mar 12 at 0:00
**Time Limit** None **Allowed Attempts** Unlimited

Take the Quiz Again

## Attempt History

| | Attempt | Time | Score |
|---|---|---|---|
| LATEST | Attempt 1 | 4,735 minutes | 4 out of 9 |

Submitted Mar 27 at 1:08

**Question 1** **1 / 1 pts**

If we set the heuristic *h(n) := 0* for all nodes *n*, what does A* become?

- Breadth-first search
- Depth-first search
- Correct! Uniform-cost search (Dijkstra)

  Correct! Same expansion order. (Details in book-keeping of open/closed states may differ)
- Iterative Deepening

## Question 2

0 / 1 pts

If we set *h(n) := 0* for all nodes *n*, what can *greedy best-first search* become?

- Breadth-first search
- Depth-first search

ou Answered

- Uniform-cost search

*h* implies no ordering of nodes at all, so this fully depends on how we break ties in the open list.
If we break ties using:
(A): FIFO -> becomes Breadth-first search
(B): LIFO -> becomes Depth-first search
(C): Order on $g$ -> becomes Uniform-cost search.
(Details in book-keeping of open/closed states may differ.)

orrect Answer

- All of the above

## Question 3

1 / 1 pts

Is *informed search* always better than *blind search*?

- True

Correct!

- False

Correct!

- In greedy best-first search, the heuristic may yield larger search spaces than uniform-cost search.
  - E.g., in path planning, say you want to go from Melbourne to Sydney, but your heuristic is telling you that is better to go through Perth: h(Perth) < h(Canberra).
- In A* with an admissible heuristic and duplicate checking, we cannot do worse than uniform-cost search:
  - h(s) > 0 can only reduce the number of states we must consider to prove optimality.
- Also, in the above example, A* doesn't expand Perth with *__any__* admissible heuristic, because the accumulated cost to Perth is much larger than the accumulated cost to Sydney: g(Perth) > g(Sydney)!

I would encourage you to draw a small graph containing 3 nodes: Melbourne, Sydney and Perth. Start the search in Melbourne, and set Sydney as your goal. Draw the costs of travelling from any pair of cities in hours: Melb-Pert = 30h, Perth-Syd=45, and Melb-Syd=10h. Test the statements above to check your intuitions.

## Question 4

**0 / 1 pts**

# Consider general search problems, which of the following are true?

**orrect Answer**

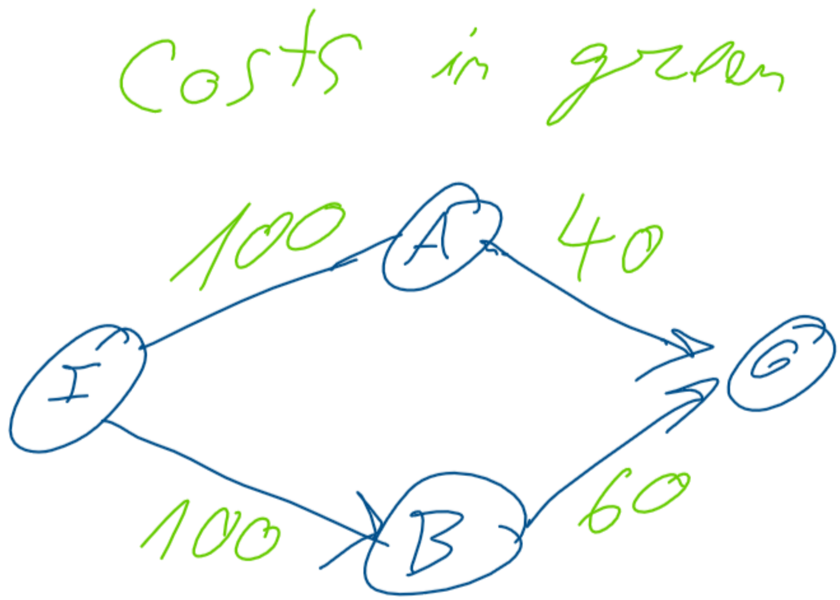
- [ ] Code that implements A* tree search can be used to run Uniform-cost search.

ou Answered

A* tree search is optimal with any heuristic function.

A* is <u>guaranteed</u> to be optimal iff the heuristic is admissible. That is, if for all nodes *n*, h(n)≤h*(n). It needs to be always lower than the true optimal heuristic.

Try to create a non-admissible heuristic that would make A* find a suboptimal solution. The initial state is I, and the goal state is G.




ou Answered

A* tree search is complete with any heuristic function.

Only if the heuristic is safe!

If the heuristic is not safe, then it can assign $h(n) := \infty$ to all nodes *n* that lead to a solution

ou Answered

A* graph search is guaranteed to expand no more nodes than DFS.

If your heuristic is misleading, there's no such guarantee. Try to build a graph where A* expands more nodes than DFS, you'll need to design a misleading heuristic.

orrect Answer

The max of two admissible heuristics is always admissible.

A heuristic that always evaluates to h(s) = 1 for non-goal search nodes s is always admissible.

## Question 5

1 / 1 pts

Heuristic functions estimate the distance from a state to:

- the closest neighbour
- the initial state

Correct!

- the closest goal state

Correct! heuristic functions estimate the cost of solving the problem from the current state, which is the cost to the nearest goal

- the furthest goal state

## Question 6

0 / 1 pts

What is a SAFE heuristic function?

ou Answered

- If a solution does not exist from state s, then h(s) = infinity

Safe heuristics ensure that no state will have a heuristic value of infinity if a solution can be found from that state. That way states with a heuristic value of infinity can safely be ignored when searching for a solution. States without a solution may have a heuristic value of infinity, but this is not required as those states can still be safely explored, even if doing so will never lead to a solution.

orrect Answer

- If a solution exists from state s, then h(s) < infinity
- If a solution does not exist from state s, the h(s) > infinity
- If a solution exists from state s, then h(s) = infinity

**Question 7** **0 / 1 pts**

A goal aware heuristic assigns h=0 to all goal states

orrect Answer

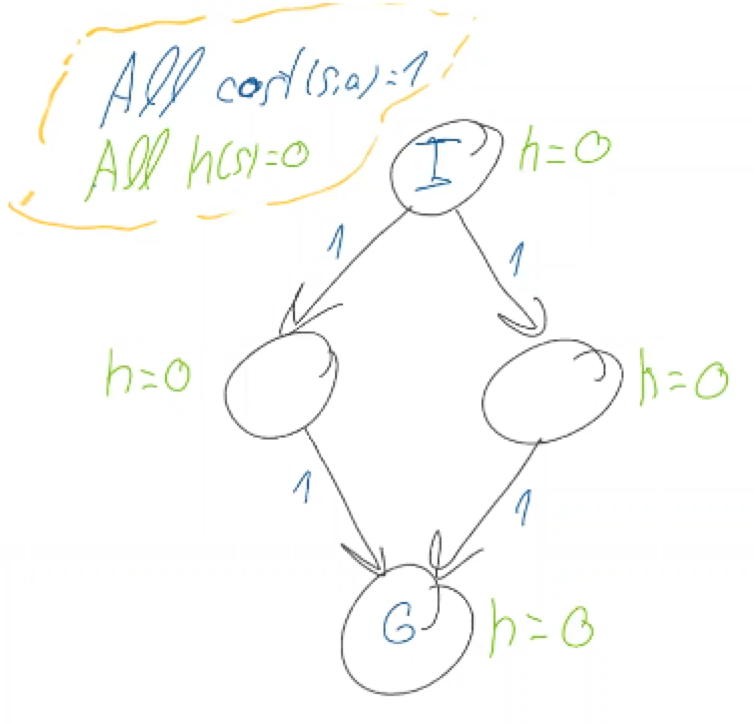
- True

ou Answered

- False

**Question 8** **0 / 1 pts**

Given the graph below, is the heuristic:




Consistent, Goal Aware and Admissible

ou Answered

Goal Aware and Admissible

Consistent: Yes because h(s) - h(s') = 0 for all transitions (as h(s)=0 for all states). c(a)=1 for all actions. Therefore h(s)-h(s') = 0 < c(a) = 1
Goal Aware: Yes because h=0 for state G
Admissible: Yes as h(s)=0 <= h*(s) for all states
Safe: Yes as no states with h*(s) = infinity

Goal Aware, Admissible and Safe

orrect Answer

Consistent, Goal Aware, Admissible and Safe

## Question 9

**1 / 1 pts**

# Heuristic search performance depends on:

Correct!

- the informedness of the heuristic and computation overhead

Correct, heuristic performance is always a balance between how well it directs the search (informedness) and how long it takes to compute (computation overhead)

- Only on how close our heuristic is to h*
- the day of the month
- only on how close our heuristic is to h=0