Eric Han
eric_han@nus.edu.sg
https://eric-han.com

Computer Science

T02 – 05 Sep 2024

Tutorial 2

CS2109s TG35,36

Content I

Administrative

1. Attendance Marking will be done at the end of the lesson via the QR code.
   - Random checks may be performed.
   - Fill Yes only when you have answered the [@] qns (different from [G])
   - Fill Yes only when you have done and showed me the bonus!
2. Survey Results
3. PS is being marked by PG tutor:
   - Comment directly on Coursemology.

Why are you taking CS2109s?

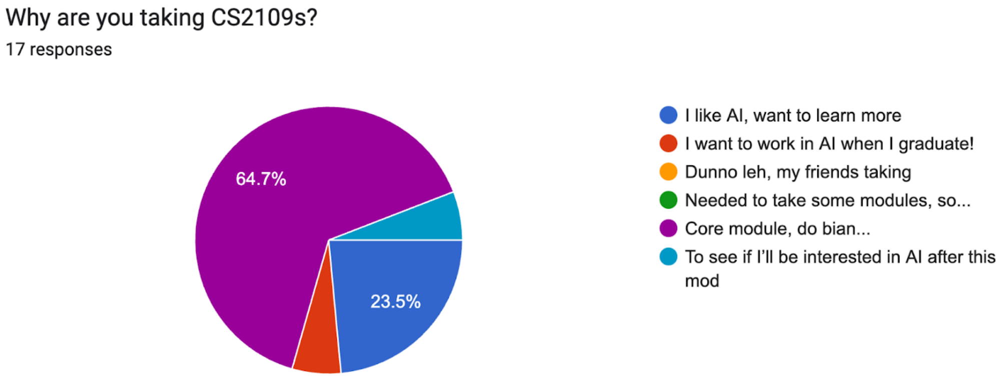


Figure 1: Core module, do bian...

How much do you know about AI/ML?

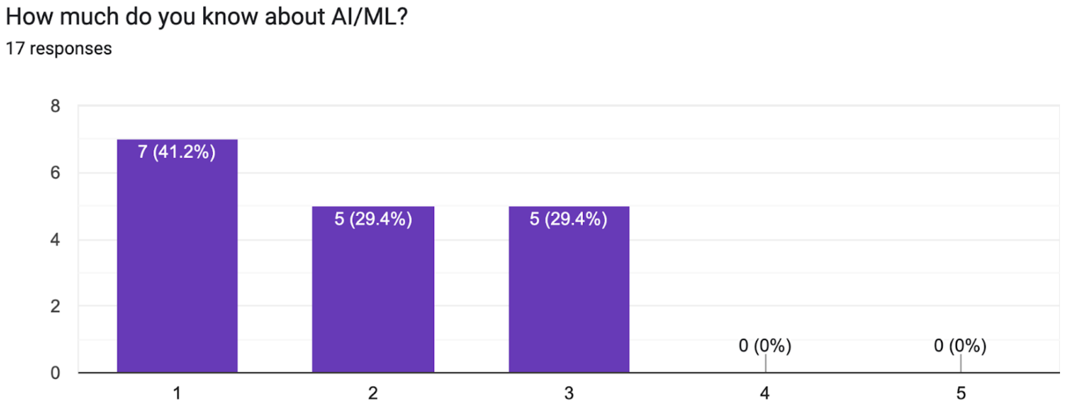


Figure 2: Most of you fall within 1-3; Hopefully we will be at 4-5.

What do you want to achieve in tutorials?

- 4x Understand the lecture content . Better prep for exams
- Clarify stuff that I'm not sure abt
- Can make through this mod on my own
- 4x Learn more
- Improve my knowledge

Any suggestions to make our classes effective?

- I think it would be really helpful if you could be very thorough (more in-depth) when explaining topics even when they are things that seem self-explanatory/pre-requisite knowledge! I know that it can be quite time consuming to do so but I think it will really help fill the gaps when it comes to understanding content! Thank you :)
- it's already effective as the way it is now

Section 1: **Pacman**

Question 1

In case you havn't: https://www.google.com/logos/2010/pacman10-i.html

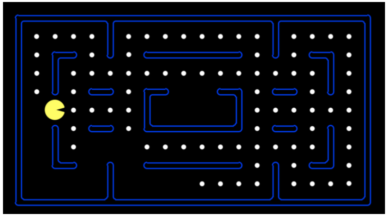


Figure 3: Pac-Man Simplified

- **State representation**: Position of Pac-Man and the positions of the uneaten pellets
- **Initial State**: Filled grid entirely with pellets
- **Goal State**: No pellets left in the grid
- **Action**: Moving up/down/left/right
- **Transition Model**: Updating the position of Pac-Man and eating pellet (if applicable)
- **Cost function**: 1 for each action taken

Give a non-trivial (whats a trivial heuristic?) admissible heuristic for this problem.

Recap

- What is the intuition behind A* Search?
- What is an admissible heuristic?
- What is a consistent heuristic?

Recap

- What is the intuition behind A* Search?
- What is an admissible heuristic?
- What is a consistent heuristic?

Summary

- A* Search: $f(N) = g(N) + h(N)$
- Admissible heuristic: $h(N) \leq h^*(N)$
- Consistent heuristic: $h(N) \leq c(N, N') + h(N')$

Where,

- $f(.)$ - Evaluation Function.
- $g(.)$ - Cost Function from start to current node.
- $h(.)$ - Estimated cost from current node to goal.
- $c(N, N')$ - Action cost from N to N'.

Answer 1

- Trivial
  - h_1 : Number of pellets left at any point in time.
- Non-Trivial
  - h_2 : The Maximum among all Manhattan distances from each remaining pellet to the current position of Pac-Man.
    - Admissible, $\max_{p \in P}$ path over all pellets ensure that $\leq h^*$. ie. furthest
  - h_3 : The average over all Euclidean distances from each remaining pellet to the current position of Pac-Man.
    - Admissible, $h_3 \leq h_2 \leq h^*$

Or.. Argue via Relaxing the problem - Pac-Man can pass through walls:

1. With less restrictions, we can take short cuts (ie. adding edges between states)
2. Optimal solution to the **original** is a solution to the relaxed, but may not be optimal
3. We can find the optimal for relaxed problem on the path by taking short cuts
4. Cost of optimal to the relaxed is always lower than the **original**'s
5. Making it an admissible heuristic for the **original** problem.

Section 2: **Route Finding**

Question 1-3 [G]

Given a graph $G = (V, E)$ where,

- each node v_n having coordinates (x_n, y_n),
- each edge (v_i, v_j) having weight equals to the distance between v_i and v_j, and
- a unique goal node v_g with coordinates (x_g, y_g),

$$h_{SLD}(n) = \sqrt{(x_n - x_g)^2 + (y_n - y_g)^2}$$

$$h_1(n) = max\{|x_n - x_g|, |y_n - y_g|\}$$

$$h_2(n) = |x_n - x_g| + |y_n - y_g|$$

1. Is $h_1(n)$ an admissible/consistent heuristic? Proof.
2. Is $h_2(n)$ an admissible heuristic? Proof.
3. Which heuristic function would you choose for A* search? And why?

Recap

- What is the intuition behind A* Search?
- How to decide admissibility?
  - If admissible... Show
  - If not admissible... Show
- What is the h^* optimal heuristic?

Think...

- How is a heuristic *useful*, how to decide which to use?
- What is a potential downfall of choosing an optimal heuristic?

Answer 1

Intuition – Proof against other heuristics: h_{SLD} and h_3, h_4:

- Admissible, proof:
  - h_{SLD} is admissible - $h^*(n) \geq h_{SLD}(n) = \sqrt{(x_n - x_g)^2 + (y_n - y_g)^2}$
    - $\sqrt{(x_n - x_g)^2 + (y_n - y_g)^2} \geq \sqrt{(x_n - x_g)^2} = |x_n - x_g|$
    - $\sqrt{(x_n - x_g)^2 + (y_n - y_g)^2} \geq \sqrt{(y_n - y_g)^2} = |y_n - y_g|$
  - $h^*(n) \geq h_{SLD}(n) = \sqrt{(x_n - x_g)^2 + (y_n - y_g)^2} \geq h_1(n) = max\{|x_n - x_g|, |y_n - y_g|\}$
- Consistency, proof:
  - Define $h_3(n) = |x_n - x_g|$, and $h_4(n) = |y_n - y_g|$.
  - Using Triangle Inequality, both $h_3(n)$ and $h_4(n)$ are consistent (similar for $h_4(n)$):

$$h_3(n) - h_3(n') = |x_n - x_g| - |x'_n - x_g| \leq |x_n - x'_n| = c(n, a, n')$$

$$h_3(n) \leq c(n, a, n') + h_3(n')$$

  - Since $h_1(n)$ is the maximum of $h_3(n)$ and $h_4(n)$, $h_1(n)$ is also consistent.

Answer 2,3

2. Not admissible, counter example:
   - Consider a graph where v_s has coordinates $(0,0)$ and v_g has coordinates $(1,1)$
   - $h^*(v_s) = \sqrt{2} < h_2(v_s) = 2$.
3. $h_{SLD}(n)$:
   - $h_2(n)$ is not admissible, so we may not get an optimal solution if we use it
   - $h_{SLD}(n) \geq h_1(n)$ so $h_{SLD}(n)$ dominates $h_1(n)$
   - will incur less search cost (on average) if we use $h_{SLD}(n)$.
   - ***Intuition**: Choose the heuristic that is closest, but under h^**

Dominance vs Admissible heuristic

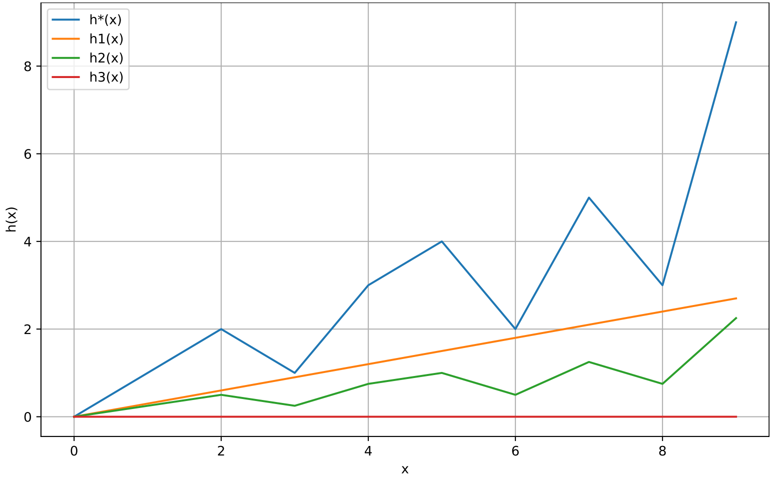


Figure 4: Dominance always better? See h_2.

Section 3: **Admissibility and Consistency**

Recap

- Admissible heuristic: $h(N) \leq h^*(N)$
- Consistent heuristic: $h(N) \leq c(N, N') + h(N')$

Question 1 [G]

Given that a heuristic h is such that $h(G) = 0$, where G is any goal state, prove that if h is consistent, then it must be admissible.

Question 1 [G]

Given that a heuristic h is such that $h(G) = 0$, where G is any goal state, prove that if h is consistent, then it must be admissible.

Answer

Intuition: Show on the number of action required to reach the goal from n to goal G.
Let the no. of actions k to be required to reach from n_k to G on optimal path $P_{n_k \to G}$.
Node n_k is k steps away from G, ie. $k = 3 \implies P_{n_3 \to G} : n_3 \to n_2 \to n_1 \to G$.

- **Base**: 1 action; i.e. node n_1 is one step away from G.
    - Since consistent, $h(n_1) \leq c(n_1, G) + h(G)$ and $h(G) = 0$
    - $\implies h(n_1) \leq c(n_1, G) = h^*(n_1) \implies$ admissible.
- **Inductive**: Assume for $k-1$ actions, path $P_{n_{k-1} \to G}, h(n_{k-1}) \leq h^*(n_{k-1})$.
    - Since consistent, $h(n_k) \leq c(n_k, n_{k-1}) + h(n_{k-1})$
    - $\implies h(n_k) \leq c(n_k, n_{k-1}) + h^*(n_{k-1}) = h^*(n_k) \implies$ admissible.

Question 2 [G]

Give an example of an admissible heuristic function that is not consistent.

Question 2 [G]

Give an example of an admissible heuristic function that is not consistent.

Answer

Intuition: Construct an admissible heuristic and make it not consistent.

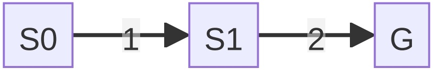


Figure 5: Example

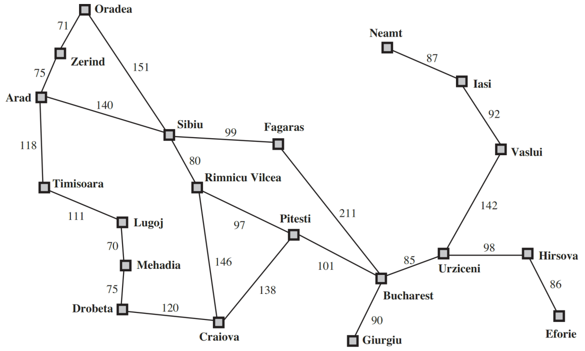
s	$S0$	$S1$	G
$h(s)$	3	1	0
$h^*(s)$	3	2	0
$h(s) \leq h^*(s)$	T	T	T

Heuristic h is

- Admissible - $\forall s : h(s) \leq h^*(s)$
- Not consistent - $3 = h(s_0) > c(s_0, s_1) + h(s_1) = 2$

Section 4: **Romania**

Question [G]



Values of h_{SLD} - straight-line distances to Bucharest

Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Figure 6: Graph of Romania.

Considering, $h(n) = |h_{SLD}(\text{Craiov}) - h_{SLD}(n)|$

1. Trace A* search (`TREE-SEARCH`) by expanding the search trees, showing (g, h, f)
2. Prove that $h(n)$ is an admissible heuristic.

Answer 1

The tuple in each node denotes (g, h, f) are shown along the trace:

```
Fagaras (0, 16, 16)
|-- Bucharest (211, 160, 371)
+-- Sibiu (99, 93, 192)
    |-- Arad (239, 206, 445)
    |-- Fagaras (198, 16, 214)
    |   |-- Bucharest (409, 160, 569)
    |   +-- Sibiu (297, 93, 390)
    |-- Oradea (250, 220, 470)
    +-- Rimnicu_Vilcea (179, 33, 212)
        |-- Craiova (325, 0, 325)
        |-- Pitesti (276, 60, 336)
        +-- Sibiu (259, 93, 352)
```

Answer 2

Let D be the straight-line distance between n and Craiova:

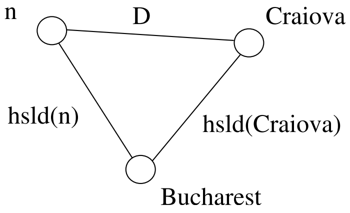


Figure 7: Triangle Inequality Example

- By the Triangle Inequality, D must be at least as much as the difference of the 2 other sides, ie. $D \geq |h_{SLD}(Craiova) - h_{SLD}(n)| = h(n)$.
- But we also know that $h^*(n) \geq D \geq h(n)$, so $h^*(n) \geq h(n)$, and
- $h(n)$ is admissible.

Note: The definition of triangle inequality is, sum of the lengths of any two sides must be greater than or equal to the the length of the third side.

Section 5: **Admissibility and Consistency**

Question

Using the example graph,

1. Show that A* using graph search returns a non-optimal solution path from start S to G when using an admissible but inconsistent $h(n)$.
2. Then, show that tree search will return the optimal solution with the same heuristic.

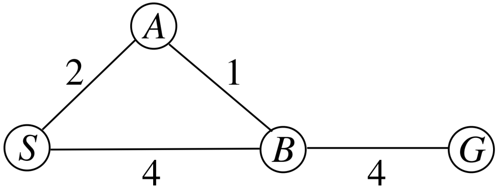


Figure 8: We assume that $h(G) = 0$.

Table 2: Inconsistent $h(S) > h(B) + 4$

$.$	S	B	A	G
$h(.)$	7	0	3	0
h^*	7	4	5	0

Answer 1

Intuition: We may miss out on some shorter paths that we mistakenly think are long after reaching the node as the heuristic function violates the triangle inequality.

The tuple in each node denotes (g, h, f) are shown along the trace:

```
S (0, 7, 7)
|--  A (2, 3, 5)
|   |-- [X] B (3, 0, 3)
|   +-- [X] S (4, 7, 11)
+--  B (4, 0, 4)
    |--  A (5, 3, 8)
    |   |-- [X] B (6, 0, 6)
    |   +-- [X] S (7, 7, 14)
    |--  G (8, 0, 8)
    +-- [X] S (8, 7, 15)
```

Frontier

```
S(7-)
B(4-S) A(5-S)
A(5-S) A(8-SB) G(8-SB)
A(8-SB) G(8-SB)
G(8-SB)
```

Answer 2

```
S (0, 7, 7)
|--  A (2, 3, 5)
|    |--  B (3, 0, 3)
|    |    |--  A (4, 3, 7)
|    |    |    |--  B (5, 0, 5)
|    |    |    |    |--  A (6, 3, 9)
|    |    |    |    |--  G (9, 0, 9)
|    |    |    |    +--  S (9, 7, 16)
|    |    |    +--  S (6, 7, 13)
|    |    |--  G (7, 0, 7)
|    |    +--  S (7, 7, 14)
|    +--  S (4, 7, 11)
+--  B (4, 0, 4)
     |--  A (5, 3, 8)
     |--  G (8, 0, 8)
     +--  S (8, 7, 15)
```

Frontier

```
S(7-)
B(4-S) A(5-S)
A(5-S) A(8-SB) G(8-SB) S(15-SB)
B(3-SA) A(8-SB) G(8-SB) S(11-SA) S(15-SB)
A(7-SAB) G(7-SAB) A(8-SB) G(8-SB) S(11-SA) S(14-SAB) S(15-SB)
B(5-SABA) G(7-SAB) A(8-SB) G(8-SB) S(11-SA) S(13-SABA) S(14-SAB)
    S(15-SB)
G(7-SAB) A(8-SB) G(8-SB) A(9-SABAB) G(9-SABAB) S(11-SA) S(13-SABA)
    S(14-SAB) S(15-SB) S(16-SABAB)
```

Section 6: **Negativity**

Question 1 [@]

Assume no negative cycles. Would A* work with negative edge weights? If yes, prove it; otherwise, provide a counterexample.

Question 1 [@]

Assume no negative cycles. Would A* work with negative edge weights? If yes, prove it; otherwise, provide a counterexample.

Answer
Recall that in the proof of A*, negative weights do not affect the optimality of A* as long as there are no negative cycles. Therefore, the same proof applies. See Tutorial solutions.

Think Further [@]

Why would A* work with negative weights not not Dijkstra's?

Question 1 [@]

Assume no negative cycles. Would A* work with negative edge weights? If yes, prove it; otherwise, provide a counterexample.

Answer

Recall that in the proof of A*, negative weights do not affect the optimality of A* as long as there are no negative cycles. Therefore, the same proof applies. See Tutorial solutions.

Think Further [@]

Why would A* work with negative weights not not Dijkstra's?

- A* considers f-score while Dijkstra's considers distance ie. g-score. The h-score gives A* additional information.

$$f(n) = g(n) + h(n)$$

Question 2 [@]

A* with negative heuristics values never over-estimates the distance to the goal? Can A* work with negative heuristics? Proof or disprove.

Question 2 [@]

A* with negative heuristics values never over-estimates the distance to the goal? Can A* work with negative heuristics? Proof or disprove.

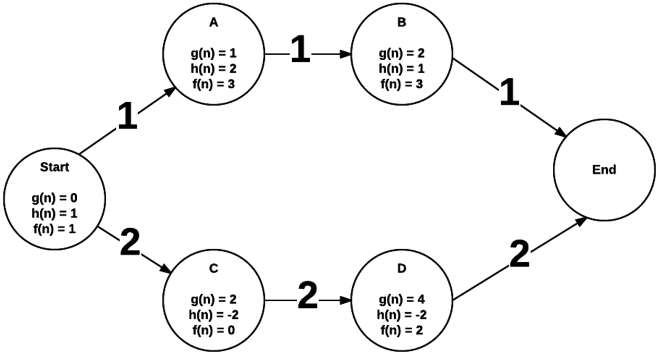


Figure 9: From StackOverflow; Negative heuristics value can break the optimality of A*.

Bonus Question

To help you further your understanding, not compulsory; Work for Snack/EXP!

Tasks

1. Fork the repository https://github.com/eric-vader/CS2109s-2425s1-bonus
2. We will be first solving **Admissibility and Inconsistency, Q1 and Q2** using code, `astar(graph, inital_node, goal_test, heuristics, is_tree, is_update)` that returns the **best path** found:
   a. Able to solve Q1 via `is_tree=False,is_update=False`
   b. Able to solve Q2 via `is_tree=True,is_update=False`
3. Some code have been implemented for you; You can reuse the PQ or use another.
4. You should print the frontier and explored at the beginning of the loop.

To claim your snack & EXP, show me your forked repository and your code's output.

Buddy Attendance Taking

1. [@] and Bonus declaration is to be done here; You should show bonus to Eric.
2. Attempted tutorial should come with proof (sketches, workings etc...)
3. Random checks may be conducted.
4. Guest student should come and inform me.

Figure 10: Buddy Attendance: https://forms.gle/jsGfFyfo9PTgWxib6

References I