



Name _____

Class _____

Date _____

What this lesson is about

Guessing what is left to pay, why the guess must never be too high, and why greedy best first is not A*.

- Admissible
- Consistent
- Why greedy best first is not A*
- Weighted A*

Questions

7 marks in all

1. What does this program print?

[1 mark]

```
import math
CELL = 5.0
goal = (34, 34)
c = (30, 20)
dx, dy = abs(c[0] - goal[0]), abs(c[1] - goal[1])
octile = CELL * (max(dx, dy) + (math.sqrt(2) - 1) * min(dx, dy))
euclid = CELL * math.hypot(dx, dy)
manhattan = CELL * (dx + dy)
print(round(octile, 1), round(euclid, 1), round(manhattan, 1))
```

2. A* runs on an eight-connected grid with the Manhattan heuristic $dx + dy$. Is that heuristic admissible? [1 mark]

- ☐ A No: a diagonal step costs $\sqrt{2}$ but Manhattan counts it as 2, so it can overestimate the remaining cost
- ☐ B Yes: Manhattan distance never exceeds the true cost on any grid
- ☐ C Yes, because Manhattan is consistent
- ☐ D No, because it is smaller than the Euclidean distance

3. What is the difference between greedy best first search and A*?

[1 mark]

- ☐ A Greedy orders the queue by h alone, dropping g , so it ignores what has already been spent and is not optimal
- ☐ B Greedy uses an admissible heuristic and A* does not
- ☐ C Greedy expands more cells than A* but finds the same path
- ☐ D Greedy uses a stack instead of a priority queue

4. Weighted A* with $w = 1.5$ is run on a map where the optimal path is 331 cm. What is the longest path, in cm, it is guaranteed never to exceed? [1 mark]

5. Which of these heuristics are admissible for A* on an eight-connected grid?

[1 mark]

Tick every answer that is true.

- ☐ A $h = 0$
- ☐ B Euclidean distance
- ☐ C Octile distance
- ☐ D Manhattan distance
- ☐ E Twice the octile distance

6. What does a consistent heuristic buy A* beyond admissibility?

[1 mark]

- ☐ A f never decreases along a path, so a node's g is final the first time it is popped and nodes never need reopening
- ☐ B A* then returns the optimal path, which an admissible heuristic does not guarantee
- ☐ C A* then expands fewer cells than greedy best first
- ☐ D A* then works with negative edge costs

7. Why is octile distance a better heuristic than Euclidean distance on an eight-connected grid?

[1 mark]

- ☐ A It is still admissible and closer to the true cost, so A* expands fewer cells
- ☐ B Euclidean distance is not admissible on a grid
- ☐ C Octile distance always equals the true cost, even with walls
- ☐ D Octile distance makes A* ignore the walls

The task: the same answer, less work

Run both searches over the same grid, eight neighbours, plain step costs of 5 cm and 5 root 2 cm, and print `cost:` (which should be identical either way), `dijkstra:` and `a_star:`, the number of cells each expanded.

```
from bugbot import *
import heapq
import math
connect()

CELL = 5.0
N = 40
INFLATE = 8.0
WALLS = [(70.0, 0.0, 8.0, 115.0), (125.0, 85.0, 8.0, 115.0)]
```

Plan your program here, then type it in and press Run.



Do it on the robot

www.bugbotlab.com/learn/u9-4-a-star/

The simulator checks it and tells you when it passes. Nothing to install, no account.

Challenges

1. Add greedy best first as a third search and print its path length and its expansions. Confirm the numbers quoted above.
2. Multiply the octile heuristic by 1.5 and by 3. Plot expansions against cost and show the curve the weight traces out.
3. Use `h = 10 * octile`. The path will be far from optimal. Work out, from the bound, how far it is allowed to be.