

Heuristics, greedy search and A*

Last week's searches were blind: they only knew how far they had come. Today we give the search a sense of direction, and prove when that sense can be trusted.

01 COLD OPEN 0-4 min

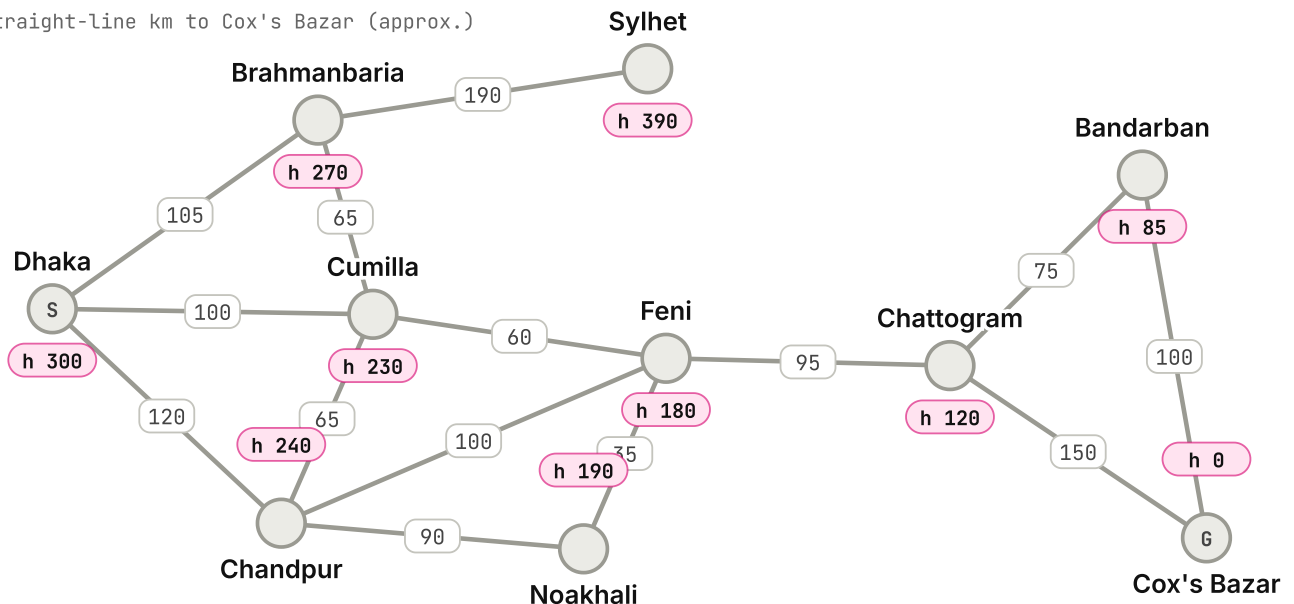
The rickshaw-puller's compass

Ask a rickshaw-puller in Old Dhaka to reach Sadarghat. No map, no GPS, but he never turns *away* from the river.

UCS last week expanded every city cheaper than the answer, in **all directions**, including Sylhet, 190 km the wrong way. It knew how far it had come, not how far it had left.

Today: add an estimate of the distance *left*, $h(n)$, and decide which city to expand using it.

h = straight-line km to Cox's Bazar (approx.)



Using h alone: **Cumilla** (230 km left). Using $g + h$: still Cumilla ($100 + 230 = 330$), but for a better reason: it is cheap to reach *and* close to the goal. The first rule is greedy search; the second is **A***.

Best-first search: one skeleton, three orders

- A **heuristic** $h(n)$ estimates the cost of the cheapest path from n to a goal. It uses knowledge about the problem; $h(\text{goal}) = 0$.
- Best-first search** is last week's UCS skeleton with the priority queue ordered by an evaluation function $f(n)$:

STRATEGY	$F(N)$	READS AS
Uniform-cost	$g(n)$	cheapest so far
Greedy best-first	$h(n)$	looks closest to the goal
A*	$g(n) + h(n)$	estimated cost of the best solution <i>through</i> n

NOTATION

$g(n)$: cost from the start to n (known).

$h(n)$: estimated cost from n to the goal (guessed).

$h^*(n)$: true cheapest cost from n to the goal (unknown during search).

★ POP CULTURE

A* was published in 1968 by Hart, Nilsson and Raphael at SRI, for **Shakey**, the first mobile robot that planned its own moves. It is older than the Apollo moon landing.

Greedy best-first: fast, short-sighted

- Always expand the node with the smallest h . On our map it walks straight down the coast: DHK, CUM, FEN, CTG, COX, **5 expansions** (UCS needed 10).
- Not optimal**: it ignores g , so a node that looks close but was expensive to reach still wins.
- Not complete** as tree search: it can oscillate between two nodes that both look close (a dead end next to the goal). Graph search in a finite space fixes that.
- Worst case $O(b^m)$ time and space, but a good h cuts this dramatically.

! GOTCHA

"Closest to the goal" is not "on the best path". A river, a cantonment wall or a one-way street can make the nearest-looking point a trap.

THINK DEEPER

When would you happily use greedy search anyway?

When a good-enough answer now beats the best answer later: real-time games, or as a fast first plan that a slower search improves.

A STRONG ANSWER MENTIONS

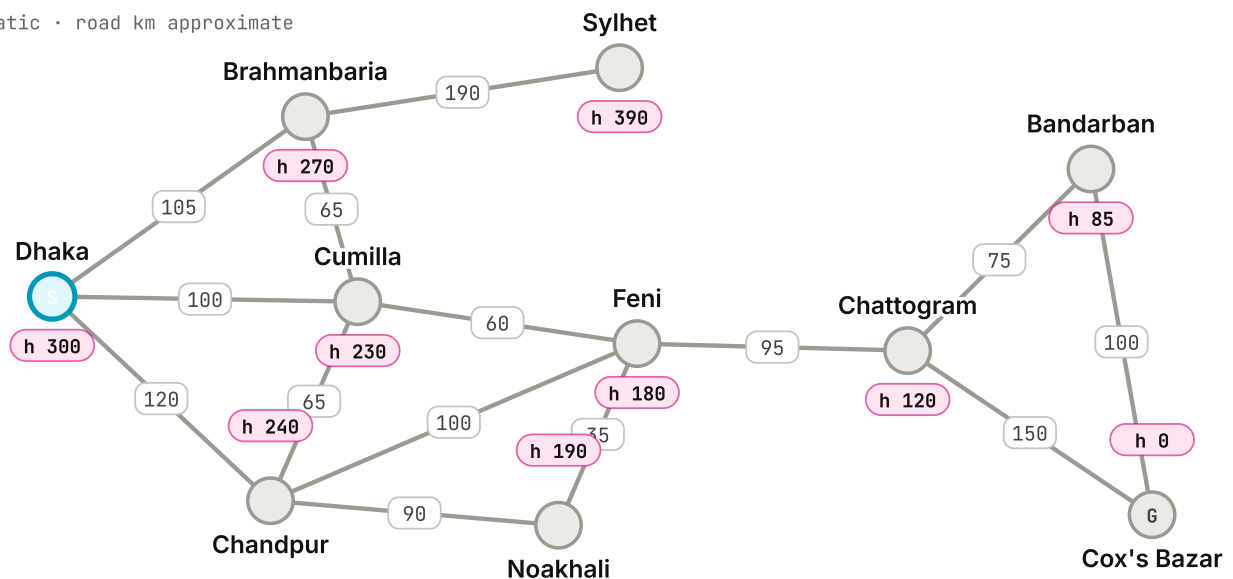
- When any reasonable solution will do and speed matters more than cost: a game NPC chasing the player, a first plan to refine later.
- When h is very accurate, greedy and A* make nearly the same choices.

04 LIVE LAB 11-16 min

Live lab: greedy vs A* on the road to Cox's Bazar

Greedy vs A*: Dhaka → Cox's Bazar The tags are $h(n)$, the straight-line km to the goal. Compare how many cities each strategy expands, and what route it returns.

schematic · road km approximate



FRONTIER

priority queue: pop lowest $f = g + h$

DHK $0+300=300$

EXPLORED

in the order expanded

empty

init Start: frontier = [Dhaka ($f = 0 + 300 = 300$)]

EXPANDED

0 / 8

COST

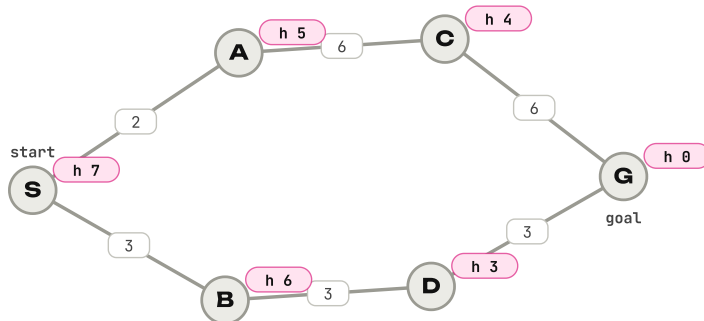
—

OPTIMAL

405 km

Your turn · 1: where greedy goes wrong

1 EXAM CQ



Edge costs on the lines, $h(n)$ in the tags. Run **greedy best-first** and then **A*** (graph search) from S to G. Give the expansion order and the path for each.

SOLUTION

1. **Greedy:** pop S ($h=7$) $\rightarrow$ A ($h=5$), B ($h=6$). Pop A $\rightarrow$ C ($h=4$). Pop C $\rightarrow$ G ($h=0$). Pop G.

$$S \rightarrow A \rightarrow C \rightarrow G = 2 + 6 + 6 = 14$$

2. **A*:** pop S ($f=7$) $\rightarrow$ A $2 + 5 = 7$, B $3 + 6 = 9$. Pop A ($f=7$) $\rightarrow$ C $8 + 4 = 12$.

3. Pop B ($f=9$) $\rightarrow$ D $6 + 3 = 9$. Pop D ($f=9$) $\rightarrow$ G $9 + 0 = 9$. Frontier: G 9, C 12. Pop G (9): goal.

$$S \rightarrow B \rightarrow D \rightarrow G = 3 + 3 + 3 = 9$$

4. A tried A first too, but once C showed $f=12$ it switched to the B branch: g kept it honest.

| Greedy expanded 4 nodes for a 14-cost path; A* expanded 5 for the optimal 9.

ANSWER

Greedy: $S \rightarrow A \rightarrow C \rightarrow G$, cost 14 (not optimal). **A*:** $S \rightarrow B \rightarrow D \rightarrow G$, cost 9 (optimal), expansion order S, A, B, D, G.

A*: cost so far + estimate to go

$$f(n) = g(n) + h(n) \approx \text{cost of the cheapest solution that passes through } n$$

- Exactly UCS with the queue on f instead of g . Goal test on **pop** (as in UCS: a goal on the frontier may still be beaten).
- If h is good, A* expands few nodes off the best path. If $h=0$, A* is UCS. If $h=h^*$, A* walks straight down the optimal path.

- Two conditions on h make A* **optimal**: admissibility (tree search) and consistency (graph search).
Next slides.

A* IN ONE PICTURE: CONTOURS

UCS spreads in circles of equal g around the start. A* spreads in ellipse-like contours of equal f , stretched towards the goal. A better h makes them thinner.



PRO TIP

The frontier entry for the goal has $f = g$ (because $h = 0$). A* stops when that is the smallest f on the frontier, i.e. when nothing else could possibly lead somewhere cheaper.

07 WORKED EXAMPLE 25-31 min

Worked example: A* by hand

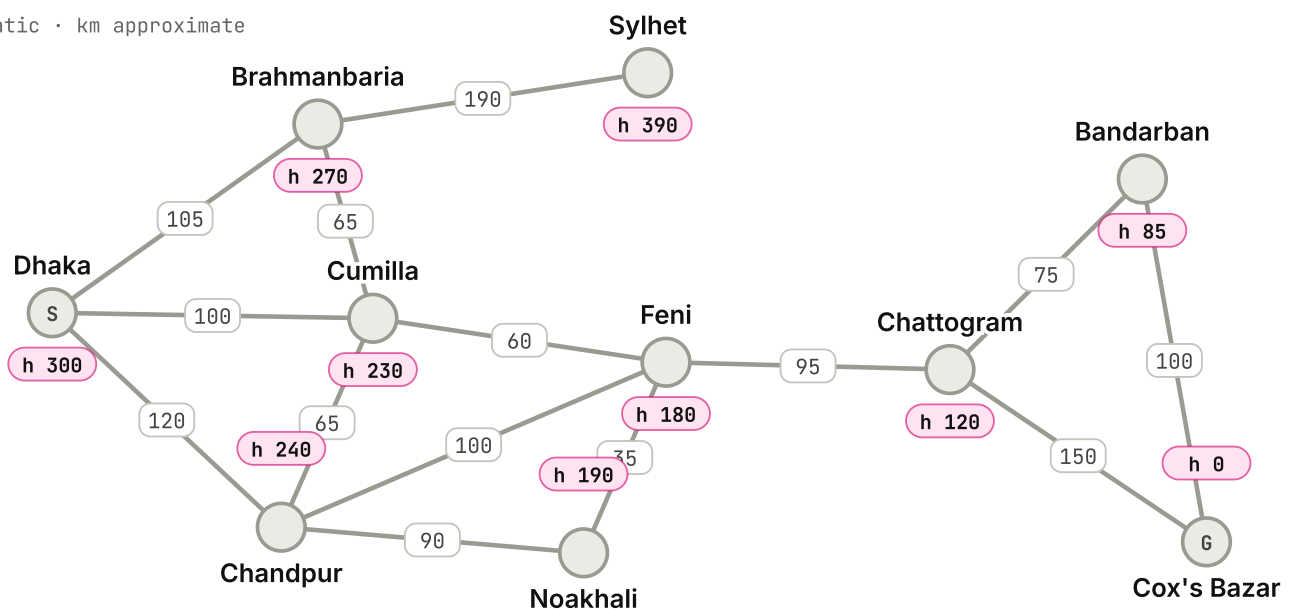
GIVEN

Last week's map with h = straight-line km to Cox's Bazar (tags). Start Dhaka, goal Cox's Bazar. Graph search.

FIND

The frontier (with $f = g + h$) after each expansion, the expansion order and the route.

schematic · km approximate



Method: A* graph search (Russell & Norvig); $f = g + h$

SOLUTION

- Pop **DHK** ($f = 0 + 300$). Add BRB $105 + 270 = \mathbf{375}$, CHP $120 + 240 = \mathbf{360}$, CUM $100 + 230 = \mathbf{330}$.

Rules: $f = g + h$, goal test on pop, ties on f go to the lower h , then alphabetical. Keep the cheaper g if a state is reached twice.

2. Pop **CUM (330)** $\rightarrow$ FEN 160 + 180 = **340**. BRB and CHP via Cumilla cost 165 > their current g : ignored.
3. Pop **FEN (340)** $\rightarrow$ CTG 255 + 120 = **375**, NOA 195 + 190 = **385**. Pop **CHP (360)**: nothing cheaper (NOA via CHP = 210 > 195).
4. Tie at 375 (CTG, BRB): CTG has the lower h . Pop **CTG** $\rightarrow$ BAN 330 + 85 = 415, **COX 405 + 0 = 405**. The goal is on the frontier, but BRB (375) and NOA (385) are still lower.
5. Pop BRB (375) $\rightarrow$ SYL 685. Pop NOA (385): nothing new. Pop **COX (405)**: goal.

DHK $\rightarrow$ CUM $\rightarrow$ FEN $\rightarrow$ CTG $\rightarrow$ COX = 405 km, 8 expansions (UCS: 10)

ANSWER

Expansion order DHK, CUM, FEN, CHP, CTG, BRB, NOA, COX (8, against UCS's 10). Route DHK $\rightarrow$ CUM $\rightarrow$ FEN $\rightarrow$ CTG $\rightarrow$ COX = 405 km, optimal.

08 **CONCEPT** 31-36 min

When can we trust A^* ? Admissible and consistent

ADMISSIBLE: NEVER OVERESTIMATES

$$h(n) \leq h^*(n) \quad \text{for every } n$$

Straight-line distance $\leq$ any road distance. Tree-search A^* with an admissible h is **optimal**.

CONSISTENT (MONOTONE): THE TRIANGLE INEQUALITY

$$h(n) \leq c(n, a, n') + h(n') \quad \text{for every edge } n \rightarrow n'$$

Consistent $\Rightarrow$ admissible, and f never decreases along a path, so graph-search A^* is **optimal** and never needs to re-open a closed node.

WHY ADMISSIBLE $\Rightarrow$ OPTIMAL (TREE SEARCH)

Let C^* be the optimal cost and G_2 a worse goal on the frontier: $f(G_2) = g(G_2) > C^*$. Some node n of the optimal path is also on the frontier, and

$$f(n) = g(n) + h(n) \leq g(n) + h^*(n) = C^* < f(G_2)$$

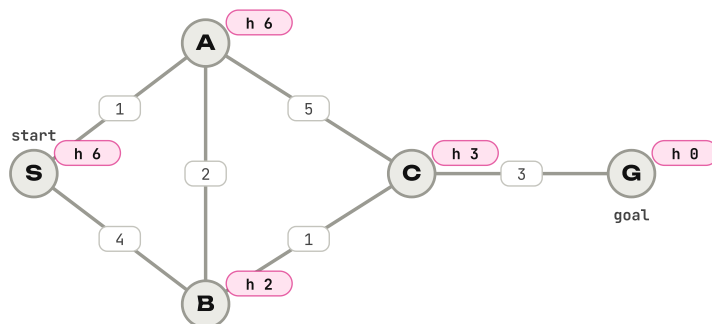
So n (and the rest of the optimal path) is popped before G_2 can be. ■

**PRO TIP**

Check consistency edge by edge: Feni–Chattogram: $180 \leq 95 + 120 = 215 \checkmark$. Bandarban–Cox's: $85 \leq 100 \checkmark$. Every edge of our map passes.

09 **YOUR TURN** 36–42 min

Your turn • 2: admissible is not always enough

2**TRAP**

(a) Is h admissible? (b) Is it consistent? (c) Run **A* graph search** (never re-expand an explored node). Is the answer optimal? Explain.

SOLUTION

1. h^* : C 3, B 4, A 6 ($A \rightarrow B \rightarrow C \rightarrow G$), S 7 ($S \rightarrow A \rightarrow B \rightarrow C \rightarrow G$). Every $h \leq h^*$: **admissible** $\checkmark$.
2. **Consistency** on A–B: $h(A) = 6 > c(A, B) + h(B) = 2 + 2 = 4 \times$. Not consistent.
3. A*: pop S (6) $\rightarrow$ A $1 + 6 = 7$, B $4 + 2 = 6$. Pop B (6) $\rightarrow$ C $5 + 3 = 8$. Pop A (7): B via A would have $g = 3$, but B is already **explored**: skipped. C via A $= 6 > 5$: no change.
4. Pop C (8) $\rightarrow$ G $8 + 0 = 8$. Pop G: cost **8**. But $S \rightarrow A \rightarrow B \rightarrow C \rightarrow G$ costs **7**.
 B was closed with $g = 4$ before its cheaper path ($g = 3$) was found. Fixes: a consistent h , or allow re-opening closed nodes (tree search / reopening).

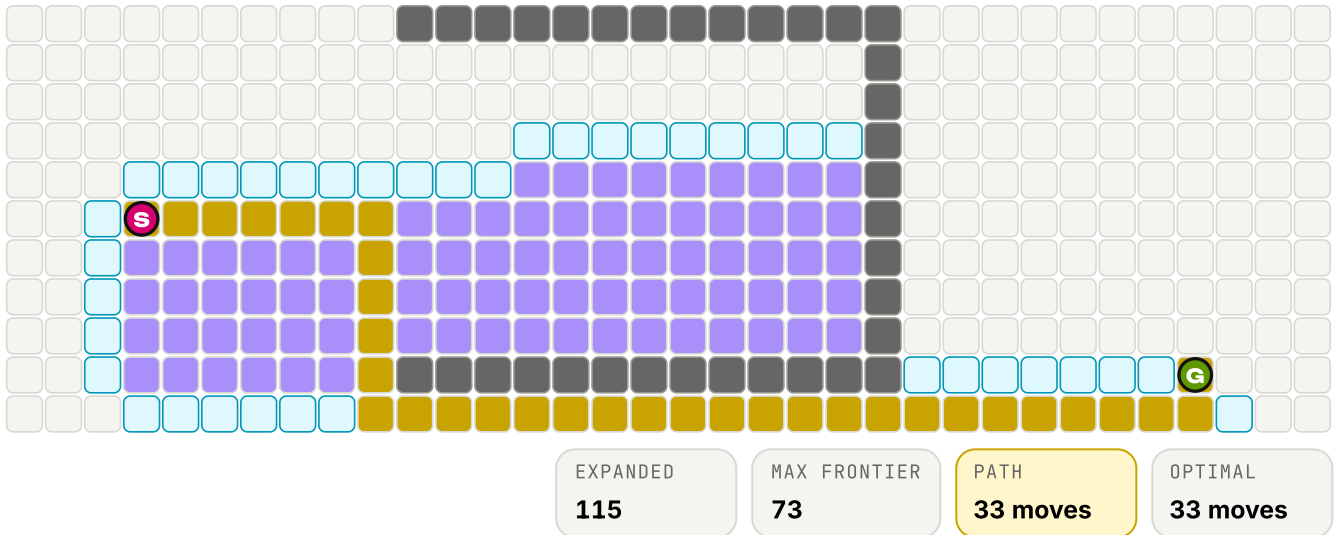
ANSWER

Admissible but not consistent (edge A–B). Graph-search A* returns $S \rightarrow B \rightarrow C \rightarrow G = 8$; the optimum is $S \rightarrow A \rightarrow B \rightarrow C \rightarrow G = 7$.

Live lab: path-finding on a grid

Path-finding on a grid: who explores what?

Each move costs 1. Purple = expanded, cyan = frontier, yellow = path. Draw walls with the mouse; drag S or G. Every run is animated.



Critical thinking • how

wrong can a heuristic be?

THINK DEEPER

Your boss multiplies the straight-line heuristic by 1.5 'to make the route planner faster'. What do you gain, and what can you lose?

Speed, at a bounded price. This is **weighted A***: with $f = g + w \cdot h$ the answer is at most w times the optimal cost. Many real planners use w between 1.2 and 2, then improve the route if time remains.

A STRONG ANSWER MENTIONS

- h is no longer admissible: A* may return a suboptimal route.
- It expands fewer nodes (greedier), usually much faster.
- For $f = g + w \cdot h$ with an admissible h , the route costs at most $w \times$ optimal: here at most 50 % worse, usually far less.

THINK DEEPER

Two admissible heuristics h_1 and h_2 give different numbers for the same node. Which should A* use?

Use **$\max(h_1, h_2)$** . It is still $\leq h^*$ (each is), and it is closer to h^* , so A* expands fewer (or equal) nodes. Next class: how to invent good heuristics.

A STRONG ANSWER MENTIONS

- Both are $\leq h^*$. The larger one is closer to the truth.
- $\max(h_1, h_2)$ is also admissible, and at least as informed as each.
- Next class: "dominance". A larger admissible h never expands more nodes.

12 **CONCEPT** 53-57 min

A*'s weakness: memory

QUESTION	GREEDY	A*
Complete?	graph search in finite spaces	yes (finite b , step costs $\geq \epsilon > 0$)
Optimal?	no	yes (admissible: tree; consistent: graph)
Time	$O(b^m)$, often far less	exponential in the error of h
Space	$O(b^m)$	keeps every generated node

! GOTCHA

A* usually runs out of **memory** long before it runs out of time: it stores the whole frontier and explored set.

MEMORY-BOUNDED RELATIVES (NAMES TO KNOW)

IDA*: iterative deepening on f instead of depth (W3's IDS idea). **RBFS** and **SMA***: forget the worst nodes when memory is full, re-create them if needed.

Real world: A* everywhere you move



GAMES

When you click a destination in a strategy game, units follow paths from A* on a grid or a navigation mesh. Engines add tricks (precomputed waypoints, path smoothing), but the core is the lab you just played with.



ROAD ROUTING

Map apps route over millions of road segments. Plain straight-line A* is too slow there; engines use A* with much better precomputed heuristics (landmarks and the triangle inequality, "ALT") or hierarchies of shortcuts. Same admissibility idea, smarter h.



HOSPITAL ROBOTS

Autonomous delivery robots carry medicines and samples along hospital corridors on an occupancy grid, re-planning with A*-family planners (D* Lite) when a trolley blocks the way.

THINK DEEPER

A Pathao rider's app uses travel time as cost and straight-line $\text{km} \div 60 \text{ km/h}$ as h. During Dhaka traffic, is this h admissible?

Yes: straight line at 60 km/h is faster than any real Dhaka trip, so it never overestimates. But it underestimates badly at rush hour, so A* is admissible yet poorly informed. A better h would use the real average speed of the city at that hour, as long as it stays an underestimate.

A STRONG ANSWER MENTIONS

- h^* is the true fastest time. Traffic makes real times longer, never shorter.
- h assumes 60 km/h in a straight line: faster than any real trip in Dhaka.
- So $h \leq h^*$: admissible (but weak: it underestimates a lot, so A* expands many nodes).

Pop quiz

1. Greedy best-first search orders its frontier by...

- A. $g(n)$
- B. $h(n)$ ✓**
- C. $g(n) + h(n)$
- D. depth

It always expands the node that *looks* closest to the goal, ignoring what it cost to get there.
Fast, but not optimal.

2. A heuristic is admissible if...

- A. $h(n) \geq h^*(n)$ for every n
- B. $h(n) \leq h^*(n)$ for every n ✓**
- C. $h(n) = 0$ at the start
- D. h decreases along every path

It never overestimates the true remaining cost $h^*(n)$. Straight-line distance is admissible for road distance: no road is shorter than the straight line.

3. A* pops a node with $f(n) = 380$. With a consistent heuristic, every node popped later has...

- A. $f \leq 380$
- B. $f \geq 380$ ✓**
- C. $g \geq 380$
- D. $h \leq 380$

Consistency makes f non-decreasing along every path, so A* pops nodes in non-decreasing f : the "contours" grow outwards.

4. $h(n) = 0$ for every n . A* then behaves exactly like...

- A. greedy search
- B. DFS
- C. uniform-cost search ✓**
- D. BFS always

$f = g + 0 = g$: UCS. (Admissible, but it knows nothing, so it expands everything cheaper than C^* .)

5. Weighted A* with $f = g + 2h$ usually...

- A. expands more nodes than A*
- B. expands fewer nodes but may return a path up to 2× optimal ✓**
- C. is always optimal
- D. never terminates

Inflating h makes the search greedier: fewer expansions, a bounded loss of optimality ($\text{cost} \leq w \cdot C^*$ when h is admissible).

15 **RECAP** 68-80 min

Three things to remember

01

Best-first search sorts by f : UCS $f = g$, greedy $f = h$ (fast, not optimal), A* $f = g + h$.

02

Admissible ($h \leq h^*$) $\Rightarrow$ tree-search A* optimal. **Consistent** ($h(n) \leq c + h(n')$) $\Rightarrow$ graph-search A* optimal, f never decreases.

03

Better $h \Rightarrow$ fewer expansions. Inflate h (weighted A^*) for speed with a bounded loss. A^* 's real limit is memory (IDA*).

HW

BEFORE NEXT CLASS

A landslide closes the **Chattogram–Cox's Bazar** road. Re-run A^* (same h) by hand from Dhaka. Which route, what cost, how many expansions? Is h still consistent?

SOLUTION

1. Same start: DHK $\rightarrow$ CUM 330, CHP 360, BRB 375. CUM $\rightarrow$ FEN 340. FEN $\rightarrow$ CTG 375, NOA 385. CHP: nothing.
2. CTG (375) $\rightarrow$ only BAN 330 + 85 = 415 now. BRB (375) $\rightarrow$ SYL 685. NOA (385): nothing.
3. BAN (415) $\rightarrow$ COX 430 + 0 = 430. Pop COX (430): goal. 9 expansions (UCS: 10).
$$\text{DHK} \rightarrow \text{CUM} \rightarrow \text{FEN} \rightarrow \text{CTG} \rightarrow \text{BAN} \rightarrow \text{COX} = 430 \text{ km}$$
4. h is still consistent: removing a road only raises h^* , and every remaining edge still satisfies $h(n) \leq c + h(n')$.

ANSWER

Route via Bandarban, 430 km, 9 expansions (DHK, CUM, FEN, CHP, CTG, BRB, NOA, BAN, COX). h stays consistent.

S. M. Sakeef Sani

Adjunct Lecturer, CSE · Northern University Bangladesh

smsakeefsani.hbitcorp.com