A* Review

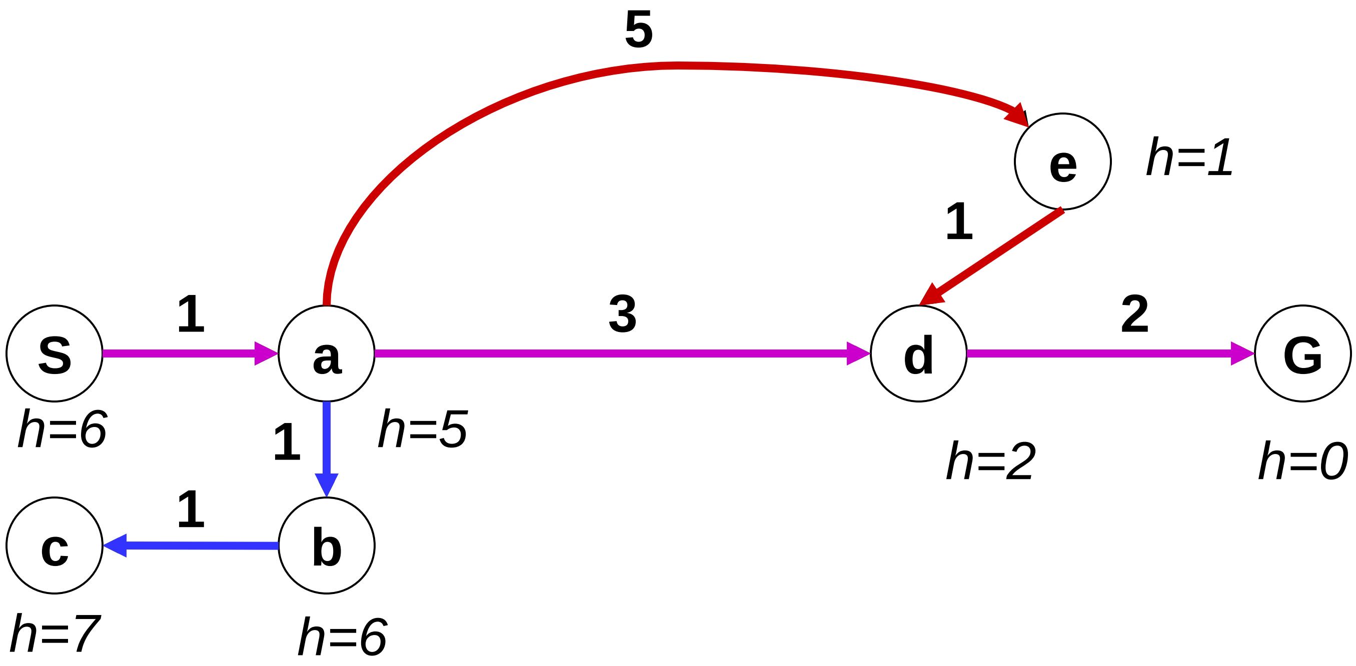
- A* uses both backward costs g and forward estimate h: $f(n) = g(n) + h(n)$

- A* tree search is optimal with admissible heuristics (optimistic future cost estimates)

- Heuristic design is key: relaxed problems can help

Combining UCS and Greedy

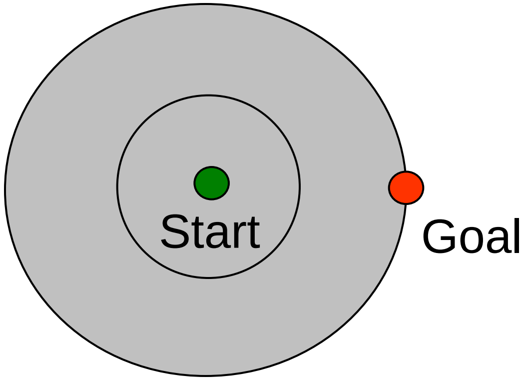
- Uniform-cost orders by path cost, or *backward cost* g(n)
- Best-first orders by goal proximity, or *forward cost* h(n)



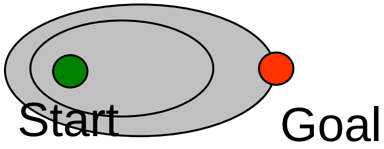
- A* Search orders by the sum: f(n) = g(n) + h(n)

UCS vs A* Contours

- Uniform-cost expanded in all directions

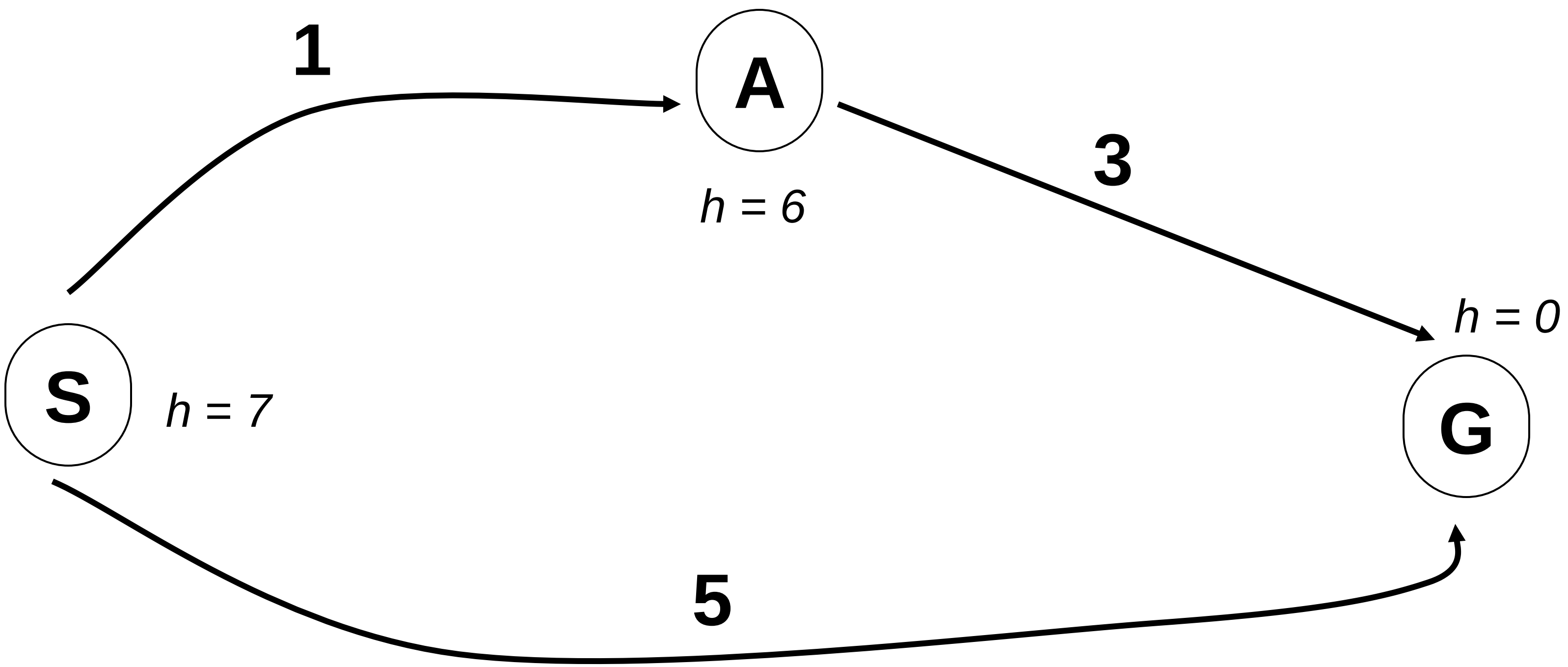


- A* expands mainly toward the goal, but does hedge its bets to ensure optimality



[

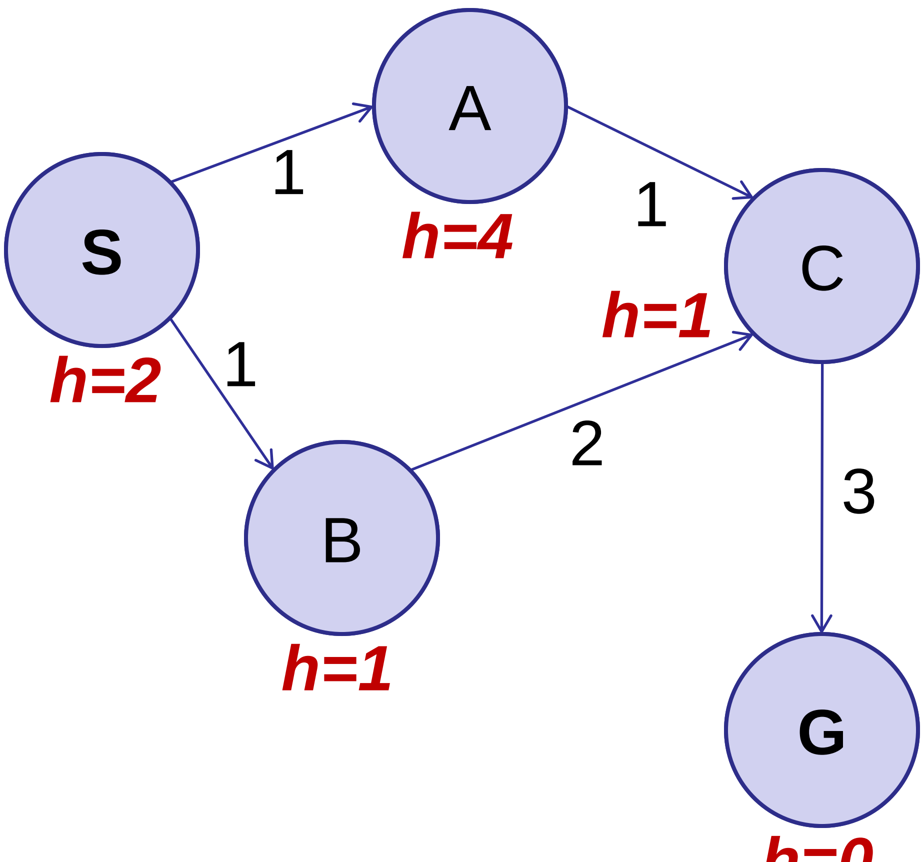
Is A* Optimal?



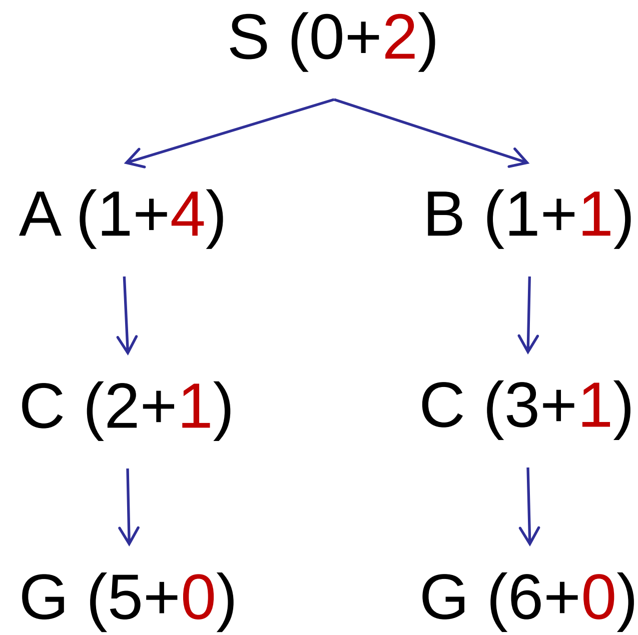
- What went wrong?
- Actual bad goal cost < estimated good goal cost
- We need estimates to be less than actual costs!

A* *Graph* Search Gone Wrong

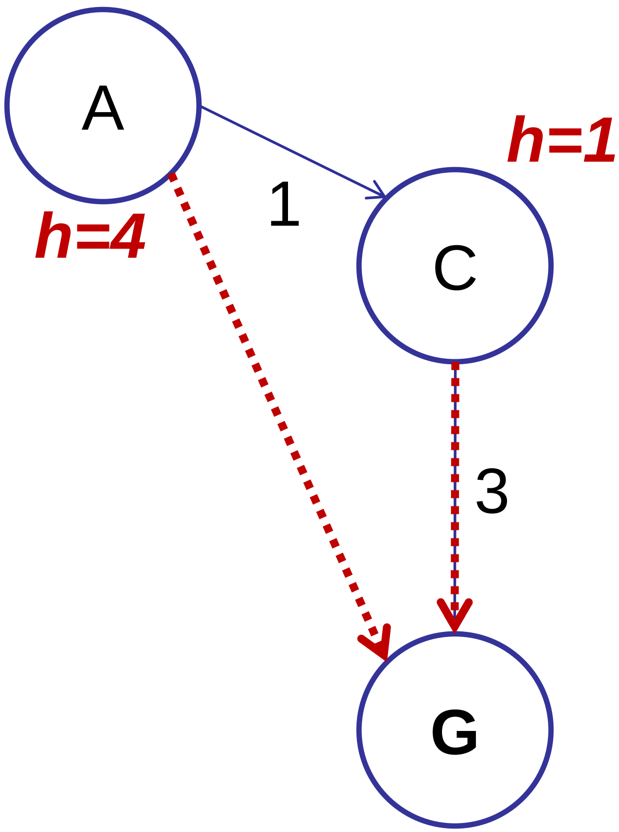
State space graph



Search tree



Consistency



The story on Consistency:

- Definition:

  cost(A to C) + h(C) ≥ h(A)

- Consequence in search tree:

  Two nodes along a path: N_{A}, N_{C}

  $g(N_C) = g(N_A) + \text{cost}(A \text{ to } C)$

  $g(N_C) + h(C) \geq g(N_A) + h(A)$

- The f value along a path never decreases
- Non-decreasing f means you're optimal to every state (not just goals)