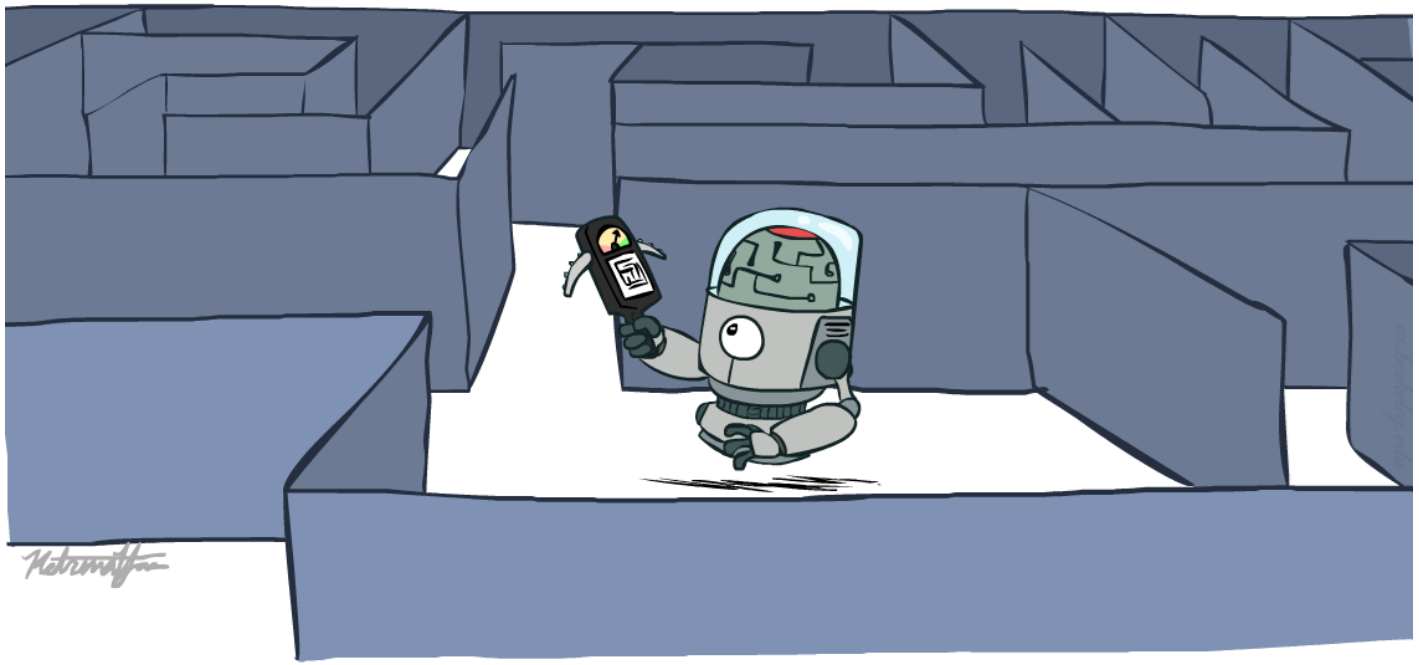
Advanced Topics in AI

Informed Search



Instructor: Prof. Dr. techn. Wolfgang Nejdl

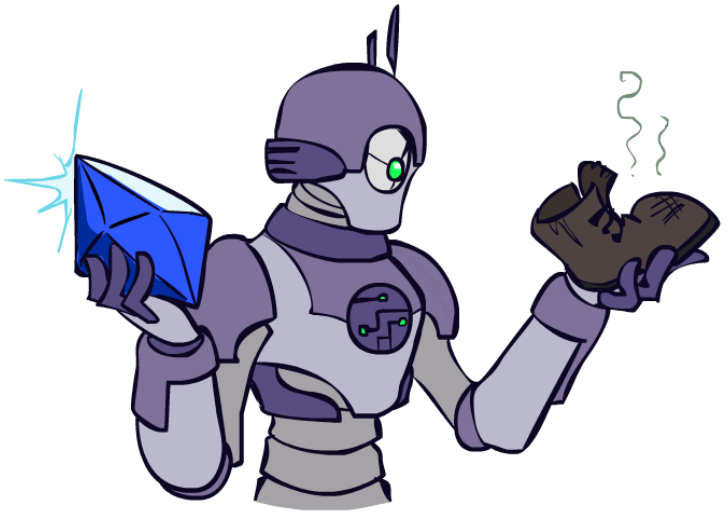
Leibniz University Hannover

Co-financed by the Connecting Europe Facility of the European Union

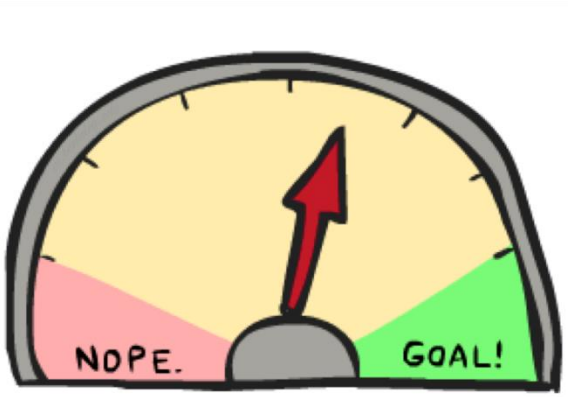
[These slides were created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley. All CS188 materials are available at http://ai.berkeley.edu.]

Informed Search

- Uninformed Search
  - Depth First Search
  - Breath First Search
  - Uniform Cost Search



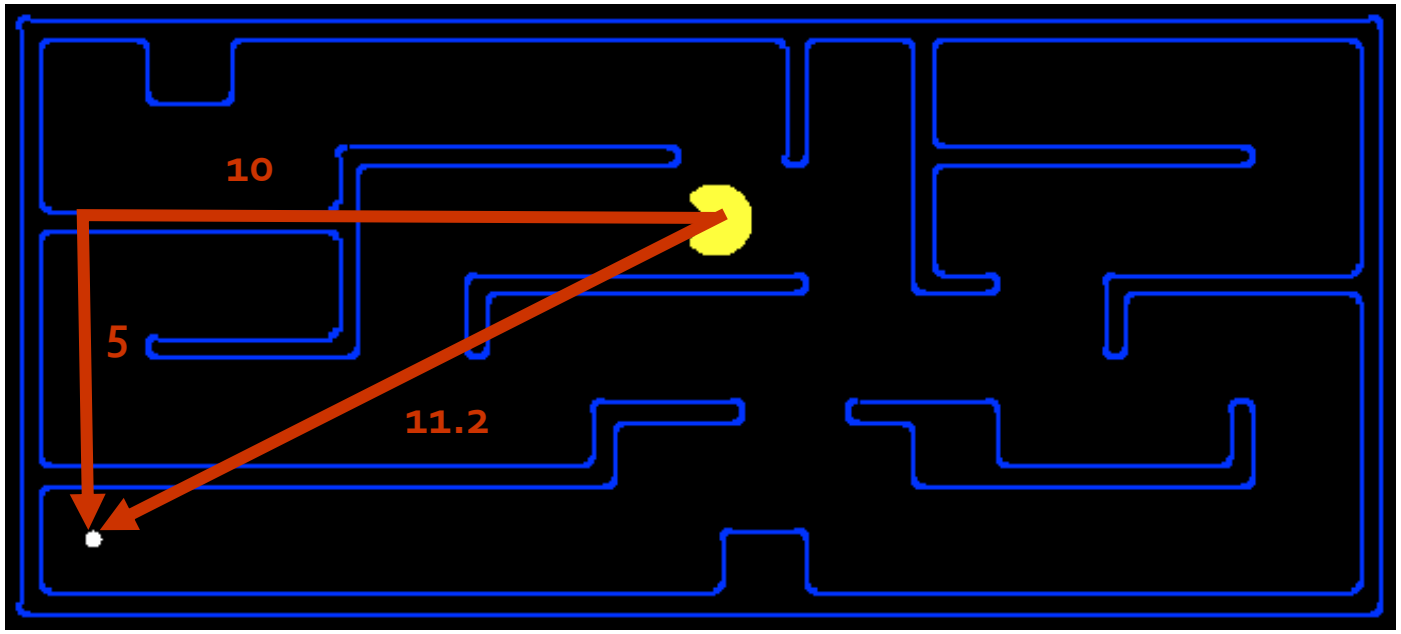
- Informed Search
  - Heuristics
  - Greedy Search
  - A* Search

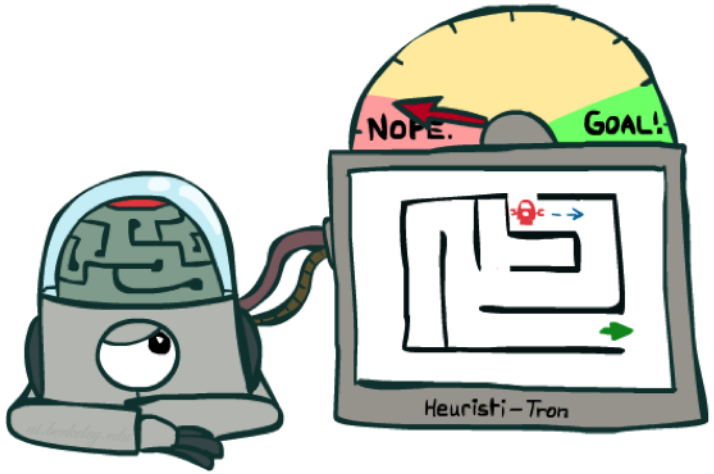


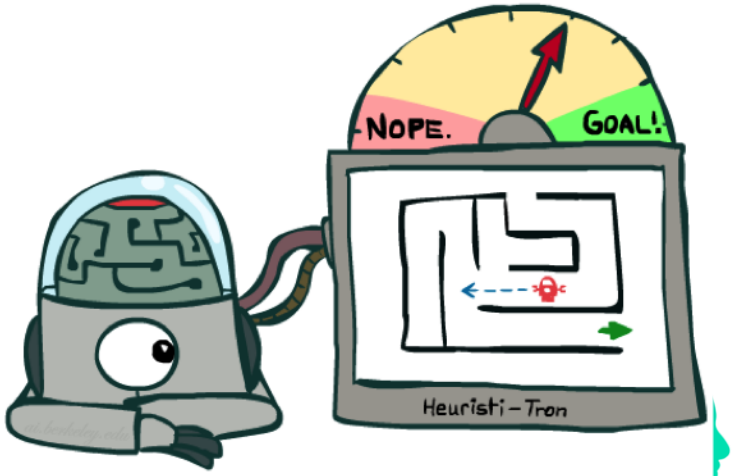
Co-financed by the Connecting Europe Facility of the European Union

Search Heuristics

- A heuristic is:
  - A function that *estimates* how close a state is to a goal
  - Designed for a particular search problem
  - Examples: Manhattan distance, Euclidean distance for pathing







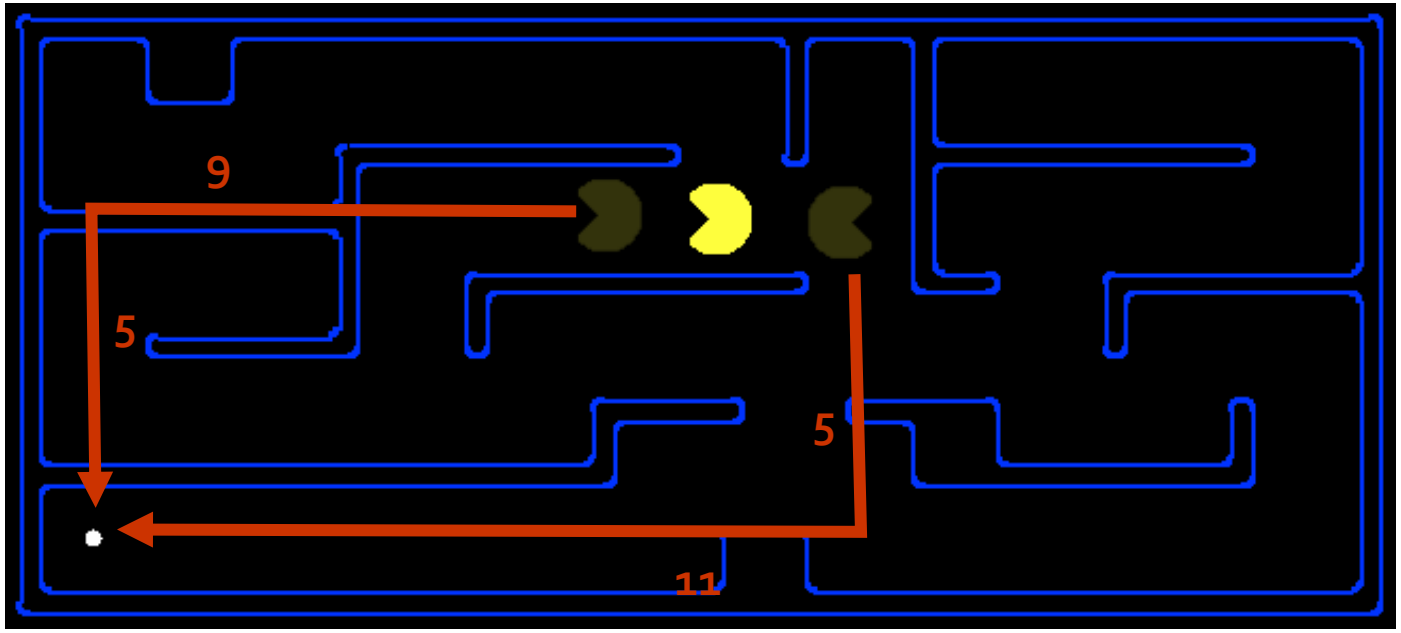
Co-financed by the Connecting Europe Facility of the European Union

Greedy Search

Co-financed by the Connecting Europe Facility of the European Union

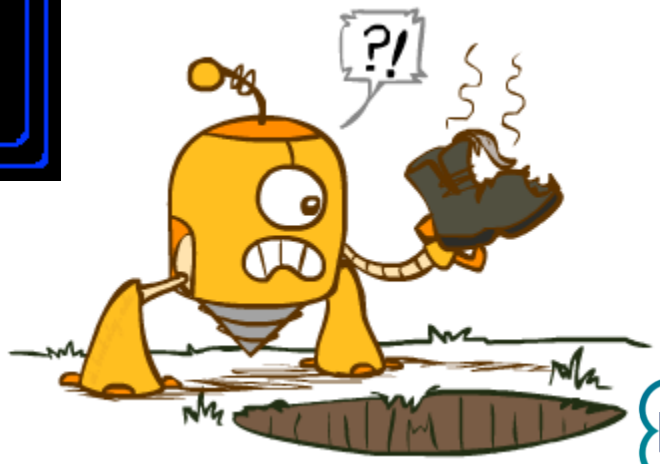
Greedy Search

- Expand the node that seems closest…



Heuristic:
Manhattan Distance

- What can go wrong?

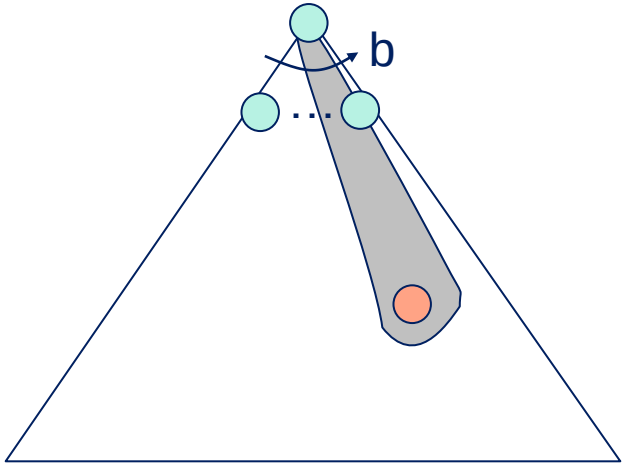


Co-financed by the Connecting Europe Facility of the European Union

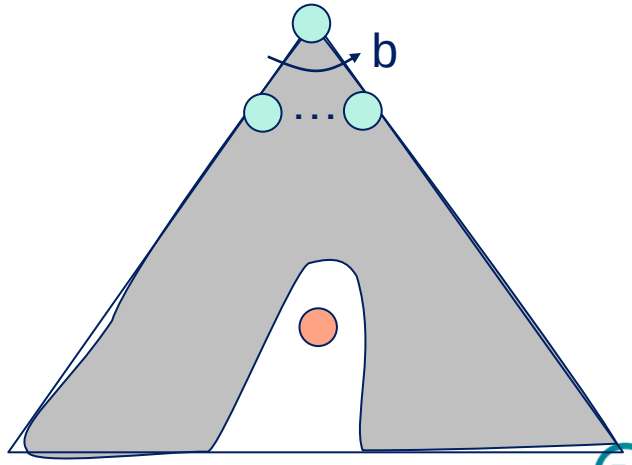
xAIM

Greedy Search

- Strategy: expand a node that you think is closest to a goal state
  - Heuristic: estimate of distance to nearest goal for each state



- A common case:
  - Best-first takes you straight to the (wrong) goal



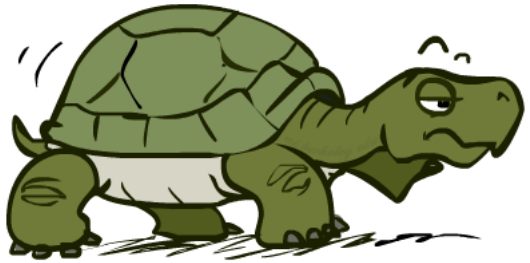
- Worst-case: like a badly-guided DFS

Co-financed by the Connecting Europe Facility of the European Union

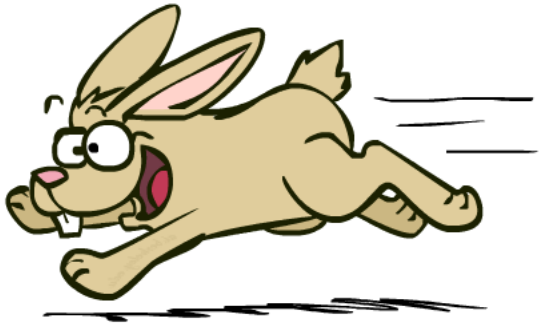
A* Search

Co-financed by the Connecting Europe Facility of the European Union

A* Search



UCS



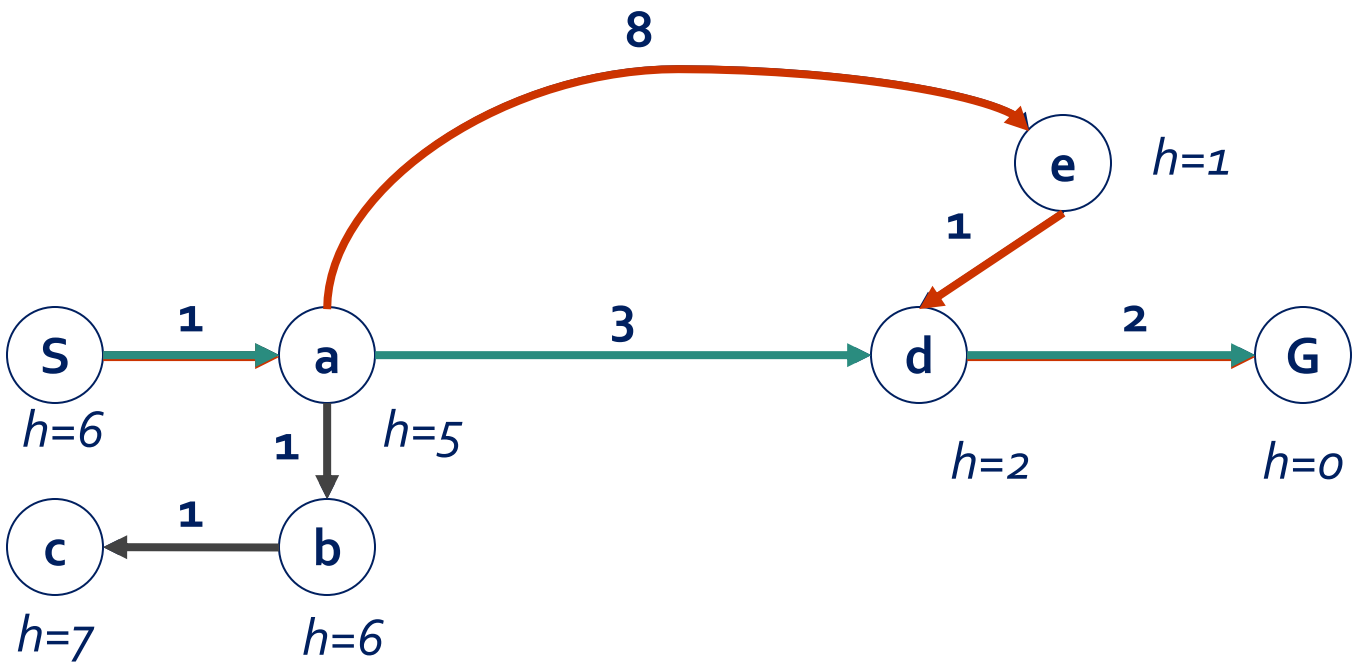
Greedy

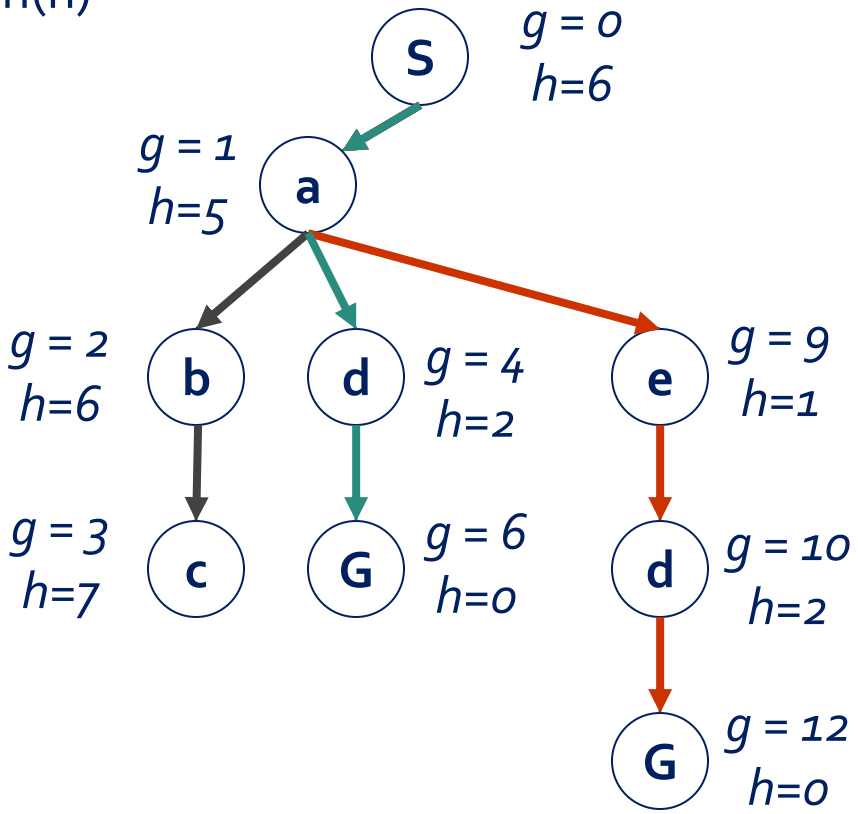
A*

Co-financed by the Connecting Europe Facility of the European Union

Combining UCS and Greedy

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





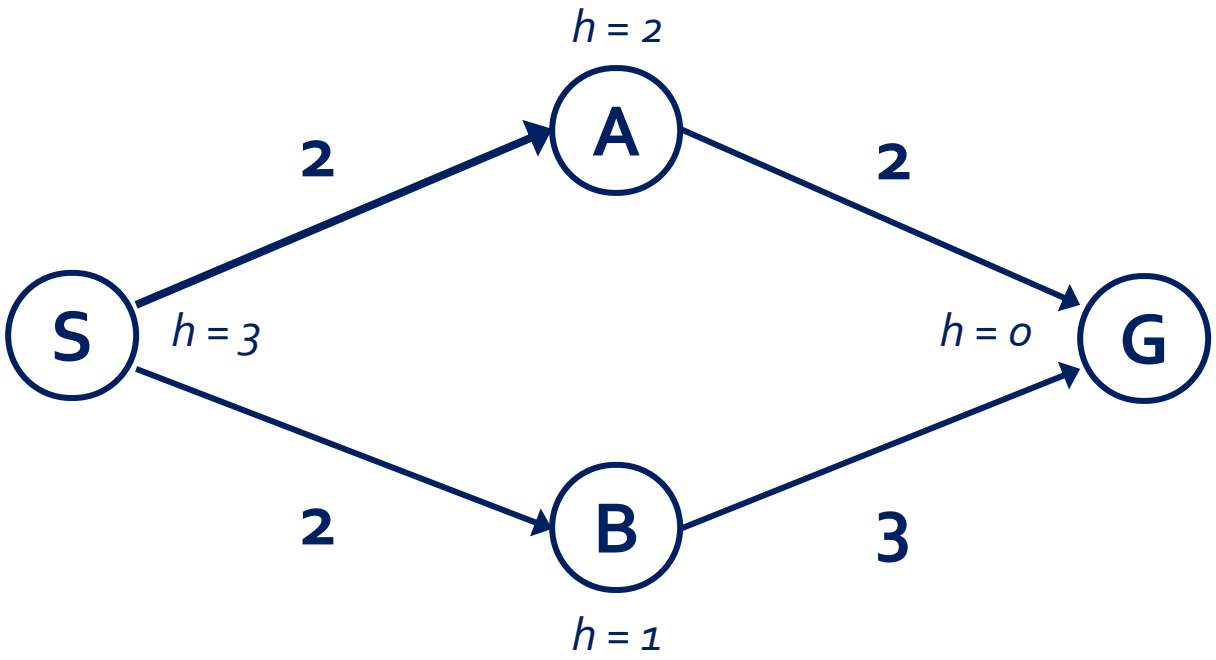
- A* Search orders by the sum: f(n) = g(n) + h(n)

Example: Teg Grenager

Co-financed by the Connecting Europe Facility of the European Union

When should A* terminate?

- Should we stop when we enqueue a goal?

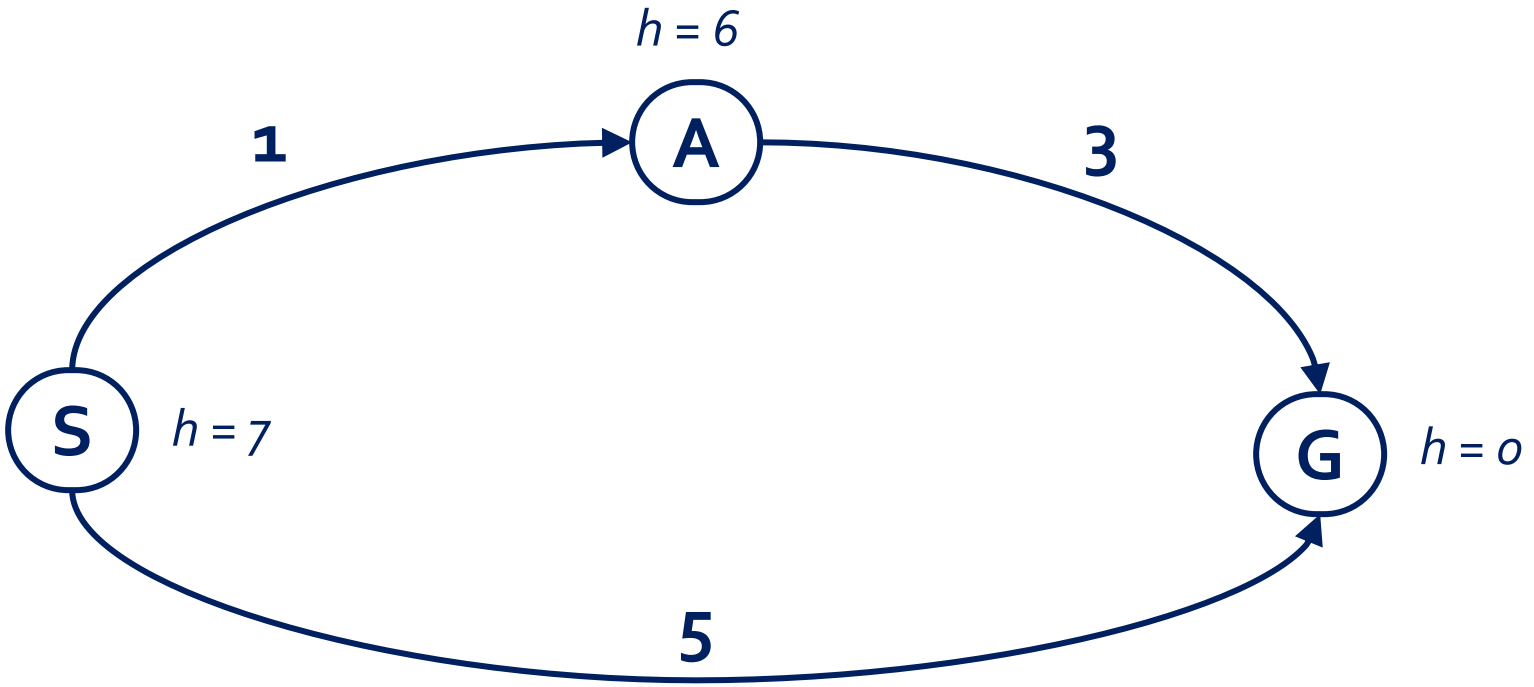


	g	h	sum
~~S~~	~~0~~	~~3~~	~~3~~
~~S → A~~	~~2~~	~~2~~	~~4~~
~~S → B~~	~~2~~	~~1~~	~~3~~
S → B → G	5	0	5
S → A → G	4	0	4

- No: only stop when we dequeue a goal

Co-financed by the Connecting Europe Facility of the European Union

Is A* Optimal?



	g	h	sum
S	0	7	7
S → A	1	6	7
S → G	5	0	5

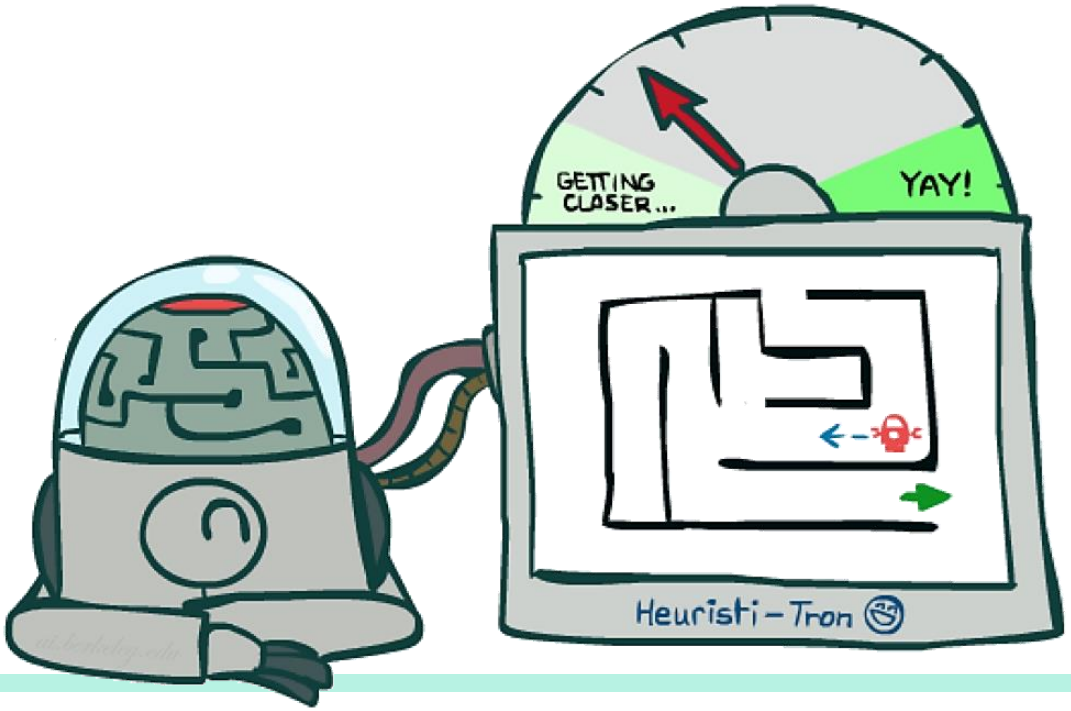
- What went wrong?
- Actual bad goal cost < estimated good goal cost
- We need estimates to be less than actual costs!

Co-financed by the Connecting Europe Facility of the European Union

xAIM

Advanced Topics in AI

Next: Heuristics



Instructor: Prof. Dr. techn. Wolfgang Nejdl

Leibniz University Hannover

Co-financed by the Connecting Europe Facility of the European Union

[These slides were created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley. All CS188 materials are available at http://ai.berkeley.edu.]