Informed search

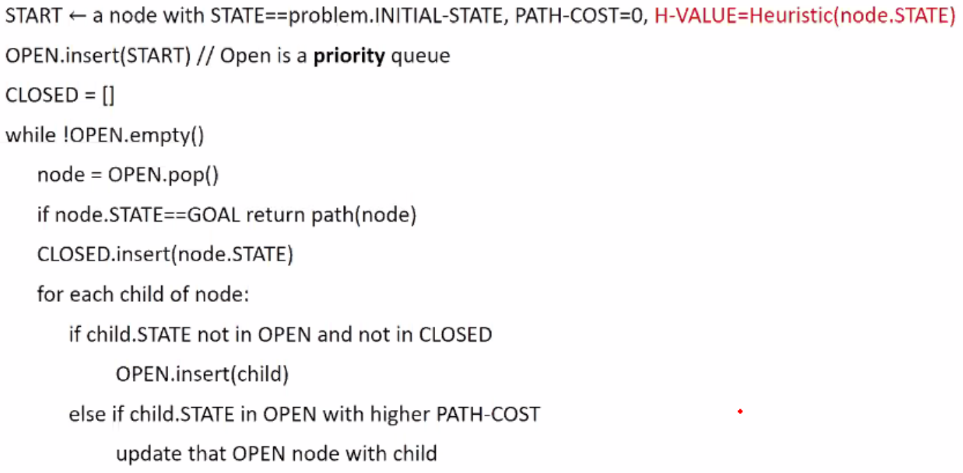
* Idea: Use an evaluation function, f(n), for each node n
  + Estimate of desirability expressed as a cost
* Open is a priority queue sorted in increasing cost

Search heuristics

* A heuristic function h(n)
  + Estimates how close the state of node n is to the goal state
  + Designed for a particular search problem
  + Eg manhattan distance, Euclidean distance, etc

Greedy (best-first) search

* Expand node that appears to be closest to goal
* Evaluation function: f(n) = h(n)
* Best-first takes you straight to (wrong) goal
* Worst case, best-first behaves like a badly-guided DFS
* O(bm) but a good heuristic can give dramatic improvement
* Keeps all nodes in memory, O(bm)

A\* search

* Guide search while avoiding paths that are already expensive
* Combines UCS and Greedy
* F(n) = g(n) + h(n)
* Optimal if estimates less than actual cost
* Complete
* Keeps all nodes in memory

Admissible Heuristics

* A heuristic is admissible if
* 0 <= h(n) <= h\*(n)
* Where h\*(n) is true cost from n to the goal ‘

Consistent heuristics

* h(n) <= c(n, a, n’) + h(n’)
* Consistent heuristics also admissible

Creating admissible heuristics

* Most of the work in solving A\* search