

# A\*, Basic theory

includes dijkstra + best first?

- \* search algo. (graph)
- \* best first
- \* finds least cost from source to dest.
- \* "A\*" → optimal for a given heuristic

A\* vs. greedy  
 \* takes 'g' into account  
 \* not greedy: not guaranteed shortest path

A\* vs. depth/breadth 1st search  
 \* pure dypn / but are special cases of A\*

A\* is breadth first  
 \* set heuristic to 0

\* Node cost  $f = g + h$   
 ↳ cost from start to node (not locally expanded)  
 ↳ heuristic from node → goal

\* h: must be admissible heuristic  
 ↳ "not over-estimate"

graceful decay of admissibility  
 ↳ overshoot: E  
 ↳ optimal path < E

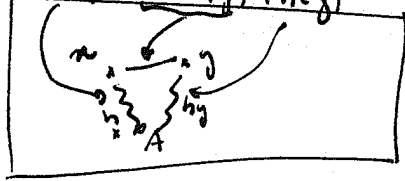
e.g. for routing: straight-line dist

↳ is surely not over-estimated!!

\* h: for closed set must be consistent (monotonic)

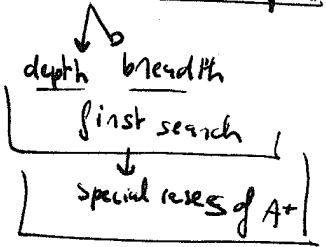
$f(x)$ 's decay +  $h(y)$

↳ this means that we cannot decrease cost by expanding to neighbor node.  
 ↳ like non-neg edge weights in dijkstra.



otherwise algo does not work!! has to search all possible

complexity (wikipedia)  
 ↳ time + memory  
 exp. ↳ poly  
 more problematic  
 ↳ IDA\* MA\* SMA\* RBFS  
 (wikipedia)



Variations  
 \* main usage: IDA\*, MA\*, SMA\*, RBFS  
 \* HPA\* ?!

- Algo:
- 1) start with init node, add to queue
  - 2) expand → find neighbors
  - 2) rem. from queue not with lowest cost
  - 3) add to neighbors
  - 4) continue until queue empty or dest. reached as lowest cost node

optimizations  
 ↳ LIFO → FIFO  
 \* ties: depth or breadth first?  
 \* keep parent no needs back-keeping / hash??