Tufts UNIVERSITY

INFORMED SEARCH 2

Fabrizio Santini | COMP 131A

VERSION 1.2

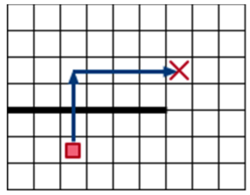
TODAY ON AI

- A* search
- Questions?

Heuristic functions are:

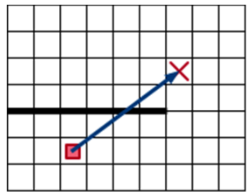
- Functions that estimate the cost of the cheapest path from a state to a goal
- Designed for the specific search problem
- $h(n) = 0$ if $n = goal$

A valid heuristic function is also include: $h(n) = 1$



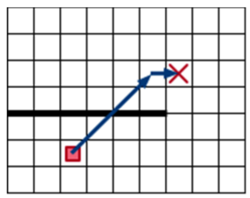
MANHATTAN DISTANCE

$h(s,g) = |s_x - g_x| + |s_y - g_y|$



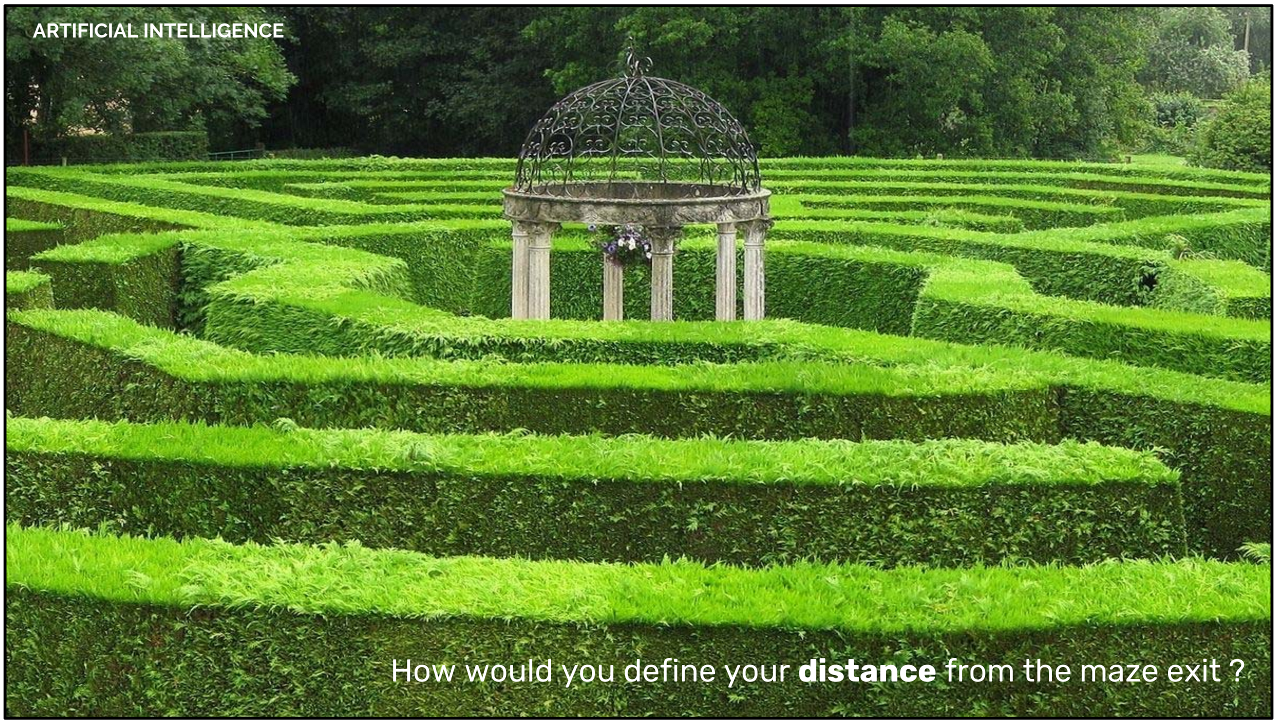
EUCLIDEAN DISTANCE

$h(s,g) = \sqrt{(s_x - g_x)^2 + (s_y - g_y)^2}$



CHEBYSHEV DISTANCE

$h(s,g) = \max(|s_x - g_x|, |s_y - g_y|)$

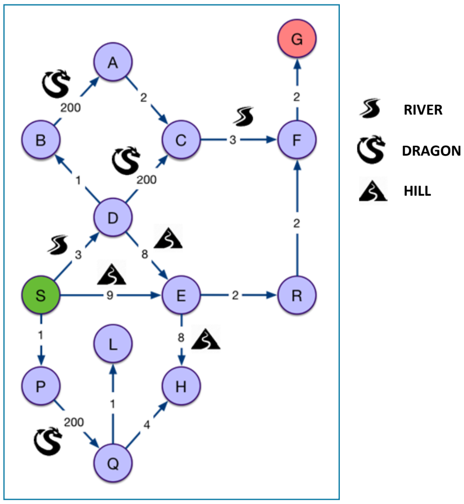

ARTIFICIAL INTELLIGENCE
How would you define your **distance** from the maze exit ?

ARTIFICIAL INTELLIGENCE

Heuristic functions

HEURISTIC FUNCTION FOR THE MAZE

MAZE GRAPH



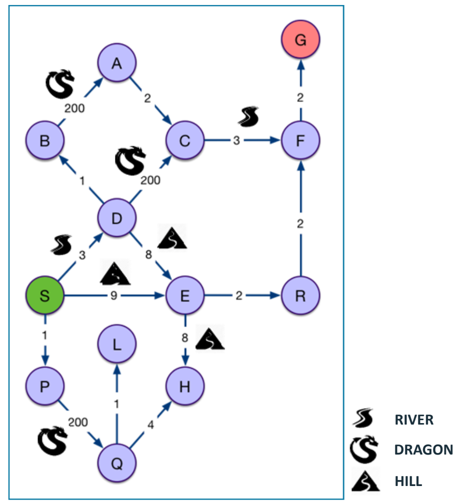
HEURISTIC FUNCTION

	A	B	C	D	E	F	G	H	L	P	R	Q	S
A		17	16	25	38	32	29	52	45	53	47	64	39
B	17		22	17	33	41	44	45	34	39	47	52	25
C	16	22		16	25	18	24	39	34	45	31	52	33
D	25	17	16		17	31	41	29	20	29	32	39	17
E	38	33	25	17		32	45	14	13	27	18	29	22
F	32	40	18	31	32		16	43	43	56	25	59	47
G	29	44	24	41	45	16		58	56	68	41	73	57
H	52	45	39	29	14	43	58		13	22	23	16	27
L	45	34	34	20	13	43	56	13		13	30	19	14
P	53	39	45	29	27	56	68	22	13		43	17	14
R	47	47	31	32	18	25	41	23	30	43		39	41
Q	64	52	52	39	29	59	73	16	19	17	39		29
S	39	25	33	17	22	47	57	27	14	14	41	29	

A* search

SECTION 01

- **Heuristic function:** Cost of the path $g(n)$ (or **backward cost**) + heuristic function to the goal $h(n)$ (or **forward cost**)
- **Node expansion:** Expand first the node that has the lowest total cost $f(n) = \min_{c} g(n) + h(c)$
- **Implementation:** Use a Priority queue, where the priority is $g(n) + h(n)$




```
function A-star(PROBLEM) return SOLUTION, or FAILURE
initialize the frontier using the initial state of PROBLEM
loop do
  if the frontier is empty then
    return FAILURE
  pop node from frontier with min_c g(c) + cost + h(c)
  if the node contains a goal state then
    return the corresponding SOLUTION
  expand the chosen node
  for each child
    if child is not in the frontier or visited then
      insert child in frontier
    else if child is in frontier with higher cost then
      replace child in frontier with child
    end
```

The algorithm behaves very similarly to a Uniform-Cost Search algorithm.

VISITED

D (3) + 41 / **44**
B (3+1) + 44 / **48**
E (9) + 45 / **54**
R (9+2) + 41 / **52**
F (9+2+2) + 16 / **29**

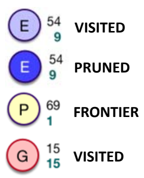
PRUNED

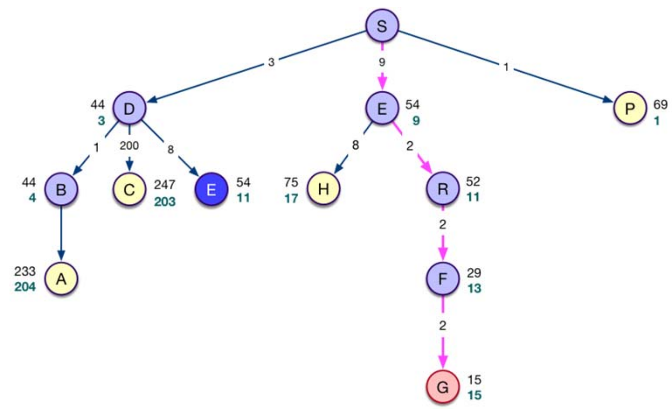
E (3+8) + 45 / **56**

FRONTIER

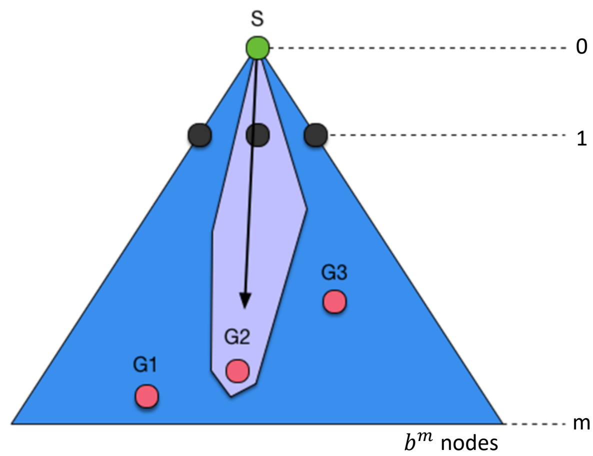
G (9+2+2+2) + 0 / **15**

P (1) + 68 / **69**
H (9+8) + 58 / **75**
A (3+1+200) + 29 / **233**
C (3+200) + 24 / **247**





- **Completeness:** Yes, only if ε is strictly positive
- **Optimality:** Yes, if the heuristic is admissible and consistent
- **Time complexity:** $O(b^m)$
- **Space complexity:** $O(b^m)$



GOOD A* is optimally efficient. It expands the minimum number of nodes

BAD Space requirements can be high

ARTIFICIAL INTELLIGENCE

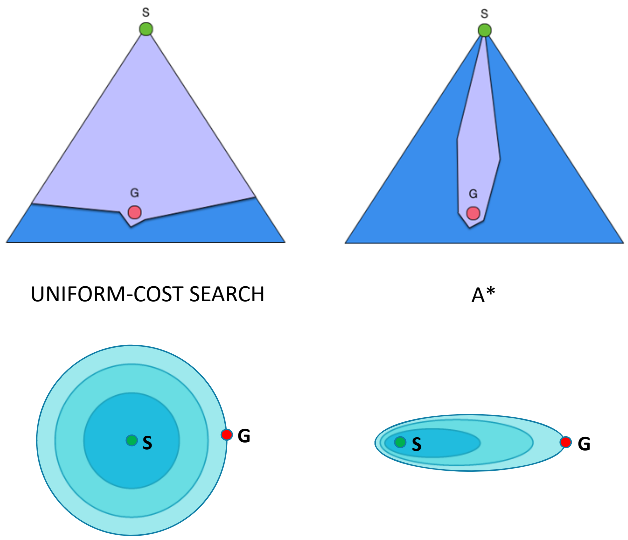
A* search

WHAT IS THE DIFFERENCE BETWEEN A* AND UCS

105

NODE EXPANSION

FRONTIER



The idea is that A* is optimal if the heuristic is **admissible**, that is:

- $0 \leq h(n) \leq h^*(n)$
- Where $h^*(n)$ is the true cost to a nearest goal. An admissible heuristic function is also called optimistic.
- Examples: Manhattan and Euclidean distances are admissible because the cost they express will always be smaller than the real cost to the goal (don't go through obstacles!)

The idea is that A* is optimal also if the heuristic is **consistent**, that is:

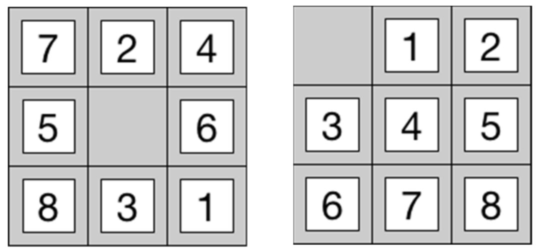
- $h(n) \leq c(n, a, n') + h(n')$

- That is, for every node n and every successor n' of n generated by any action a, the estimated cost of reaching the goal from n is no greater than the step cost of getting n' plus the estimated cost of reaching the goal from n'.

- Given two admissible heuristics $h_1(n)$ and $h_2(n)$: If $h_2(n) \geq h_1(n)$ for all nodes n, then h_2 **dominates** h_1
- It means that with h_2, A* expands fewer states, and therefore more efficient

8-PUZZLE

- $h_1(\text{board})$: number of misplaced tiles
- $h_2(\text{board})$: sum of Manhattan distances between desired and actual location of each tile



$h_1(\text{board})$ = 8
$h_2(\text{board})$ = 3+1+2+2+2+3+3+2 = 18

We want to demonstrate that, if h is **admissible**, the solution found by A* is optimal.

Let's have two goals A and B. A is optimal, B is sub-optimal.

$$g(A) < g(B)$$

We can demonstrate that there would be a contradiction if A* returned B instead of A.
Note that when selecting a node from the frontier, we select the node s such that:

$$g(s) + h(s) \leq g(s') + h(s')$$

for all other nodes s' in the frontier. This also means that selecting B:

$$g(B) + h(B) \leq g(s') + h(s')$$

Because B is a goal, $h(B) = 0$, by the definition of heuristic:

$$g(B) \leq g(s') + h(s')$$

Now, we note that A must have had some ancestor node in the frontier; let's call it n. Since h is admissible and underestimates the cost the goal:

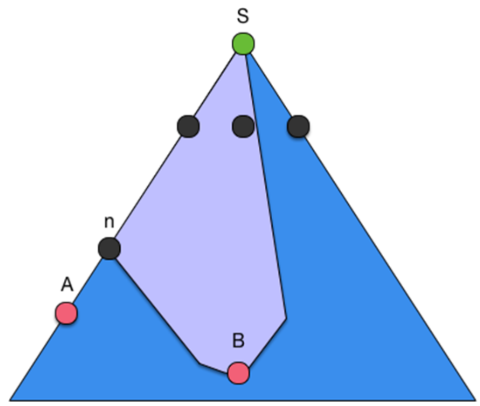
$$g(n) + h(n) \leq g(A)$$

But since B was selected instead of A, it must be that:

$$g(B) + h(B) = g(B) \leq h(n) + g(n) \leq g(A)$$

$$g(B) \leq g(A)$$

which is in contradiction with the initial assumptions. ■



- Iterative deepening A* (IDA*)

- Memory-bound A* (MA*) and Simplified Memory-bound A* (SMA*)

QUESTIONS ?