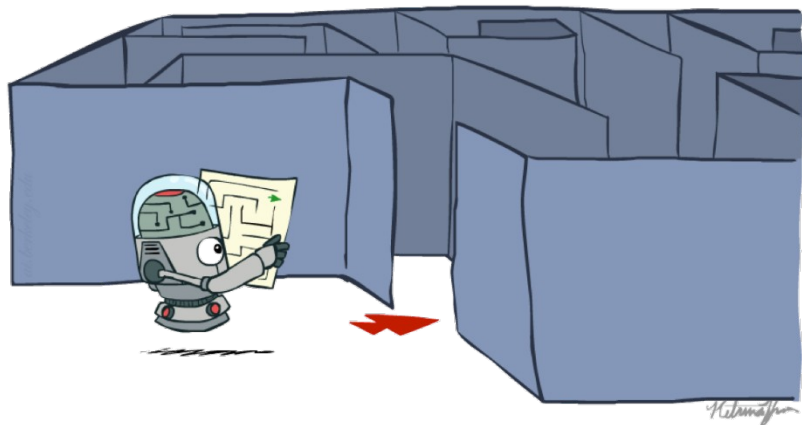


3.5 有信息（启发式）搜索策略

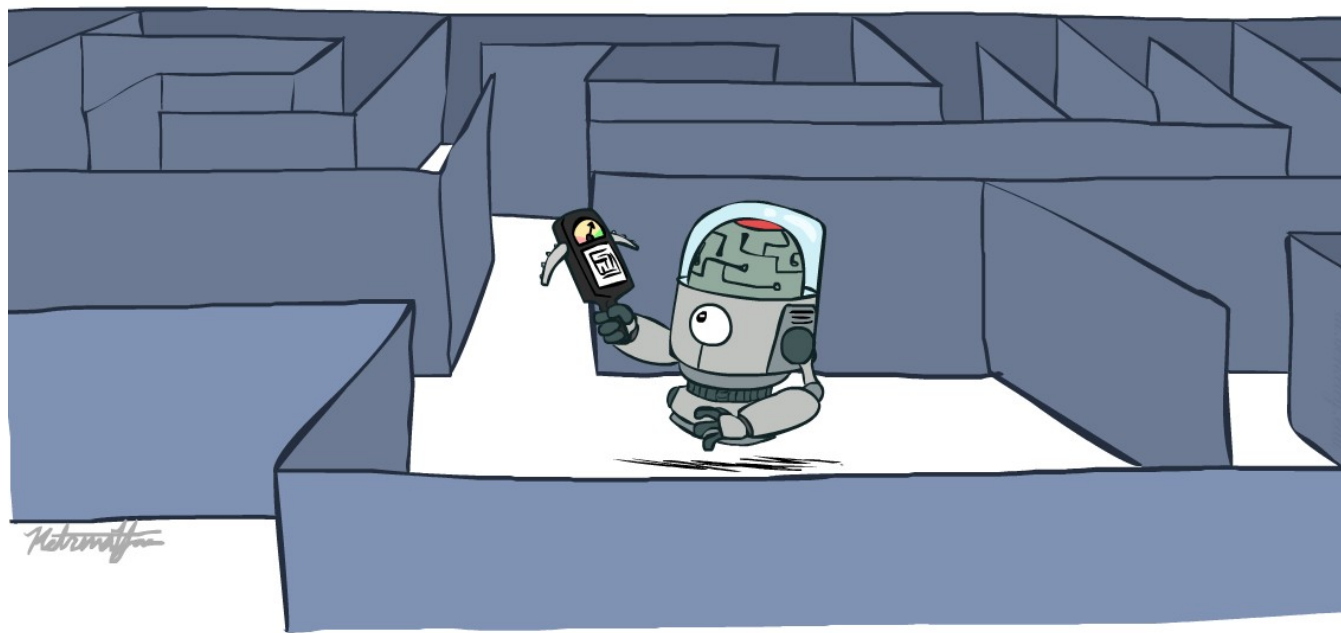


无信息搜索

（除了问题本身外，没有任何额外的信息）

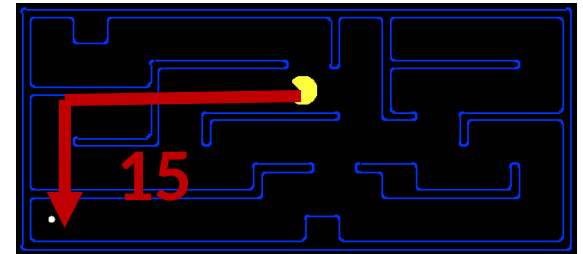
有信息搜索

（除了问题本身外，还有启发式信息）



可采纳启发式

- 若对每个结点 n , 满足 $0 \leq h(n) \leq h^*(n)$, 则 $h(n)$ 是可采纳的。
 - 其中 $h^*(n)$ 是从结点 n 到达目标结点的真实代价。
- 可采纳启发式不会过高估计到达目标的代价。
 - 例如 : $h_{SLD}(n)$ (不会高估真实距离)



4



- 定理 : 如果 $h(n)$ 是可采纳的 , A^* 搜索树搜索算法是最优的。

A* 搜索分析

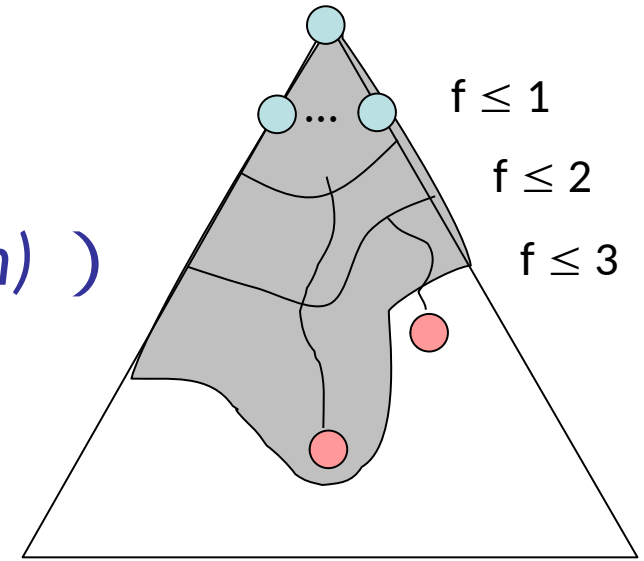
保证找到最短路径（最优解）的条件，**关键在于评估函数 $f(n)$ 的选取（或者说 $h(n)$ 的选取）。**

以 $h^*(n)$ 表达状态 n 到目标状态的真实代价，那么 $h(n)$ 的选取有如下四种情况：

- 1. 如果 $h(n)=0$ ，**一致代价搜索**，能得到最优解
- 2. 如果 $h(n)=h^*(n)$ ，搜索将严格沿着最优解路径进行，此时的**搜索效率是最高的**。
- 3. 如果 $0<h(n)<h^*(n)$ ，搜索的点数多，搜索范围大，能**得到最优解**。
- 4. 如果 $h(n)>h^*(n)$ ，搜索的点数少，搜索范围小，但不能保证得到最优解。

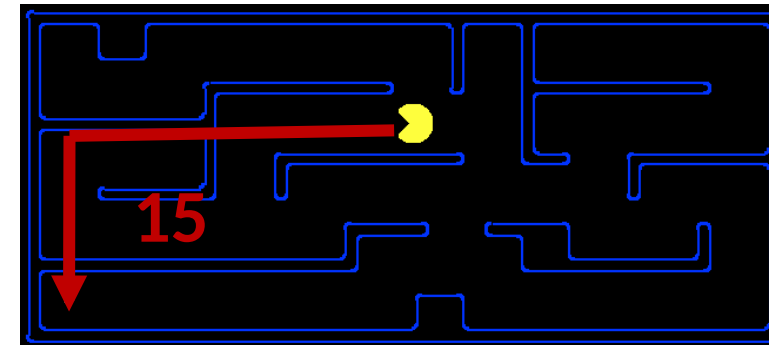
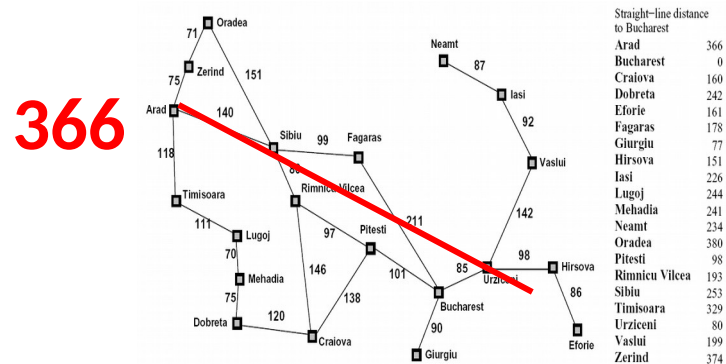
A* 搜索的性能

- 完备性 ? 是 (if finite nodes)
- 最佳性 ? 是 (if $h(n)$ 是可采纳的 : $0 \leq h(n) \leq h^*(n)$)
- 时间复杂度 ? 指数级
- 空间复杂度 ? 指数级 Keeps all nodes in memory



3.6 启发式函数

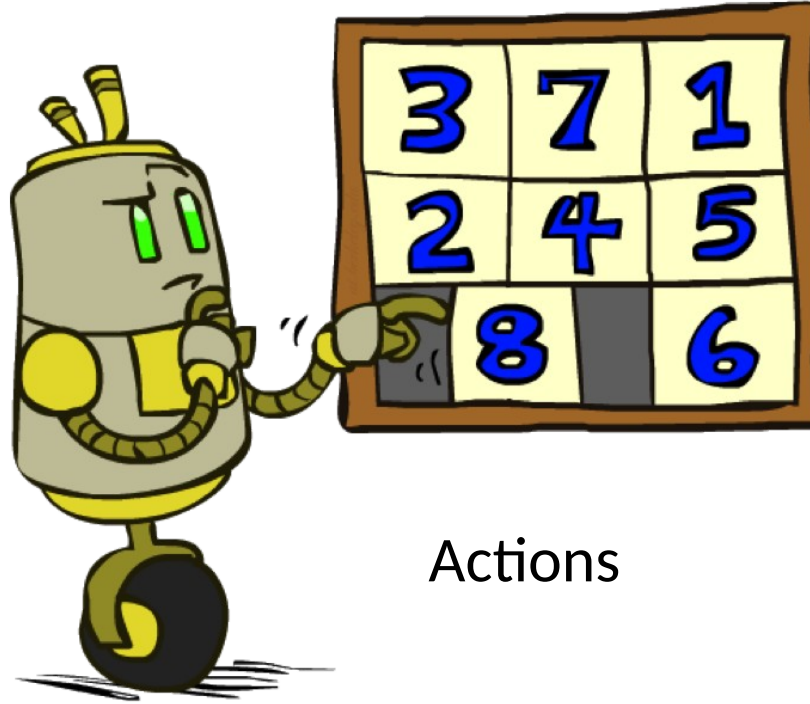
- Most of the work in solving hard search problems optimally is in coming up with admissible heuristics
- Often, admissible heuristics are solutions to *relaxed problems*, where new actions are available



Example: 8 Puzzle

7	2	4
5		6
8	3	1

Start State



Actions

	1	2
3	4	5
6	7	8

Goal State

- What are the states?
- How many states?
- What are the actions?
- How many successors from the start state?
- What should the costs be?

可采纳的启发式

E.g., 针对八数码问题，有两个常用的可采纳的启发式函数：

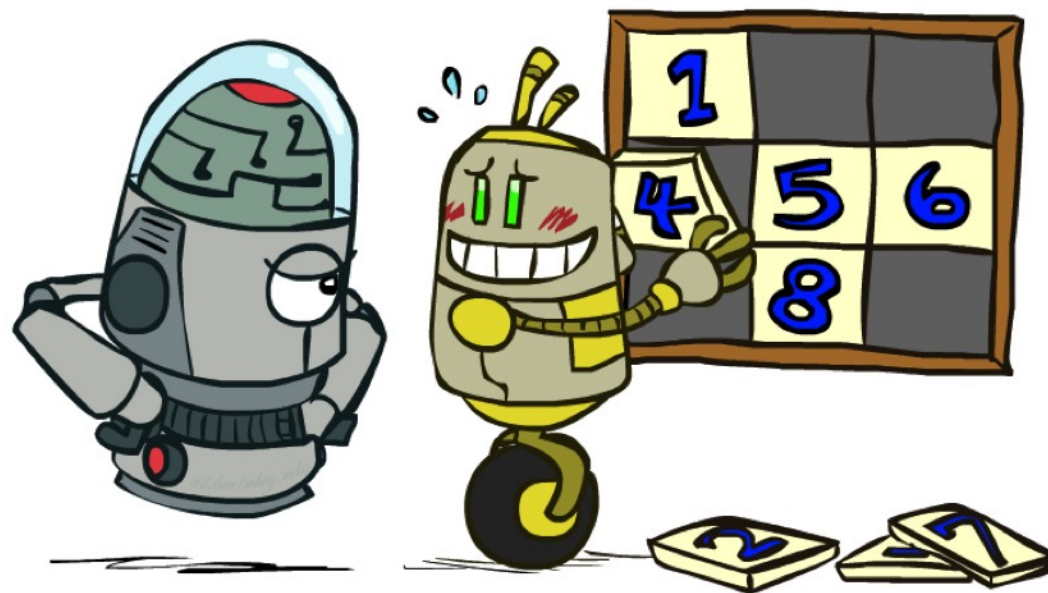
- $h_1(n)$ = 不在位的棋子数
- $h_2(n)$ = 所有棋子到其目标位置的距离和

7	2	4
5		6
8	3	1

Start State

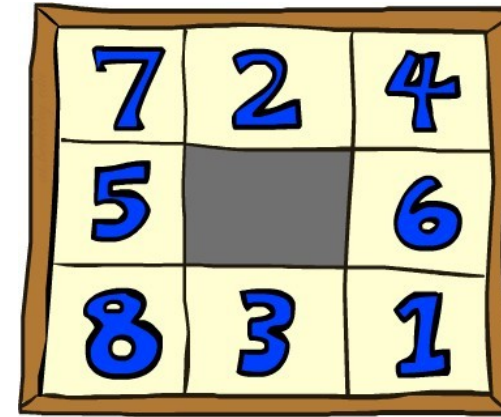
	1	2
3	4	5
6	7	8

Goal State

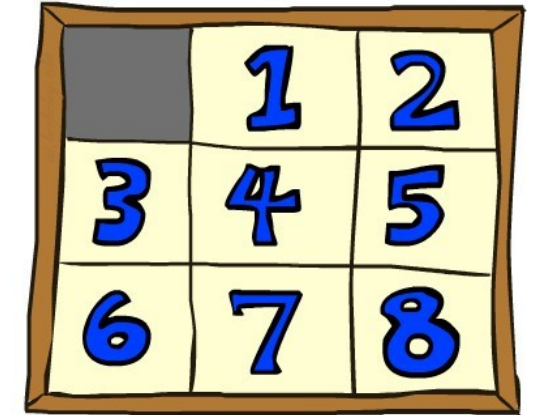


8 Puzzle I

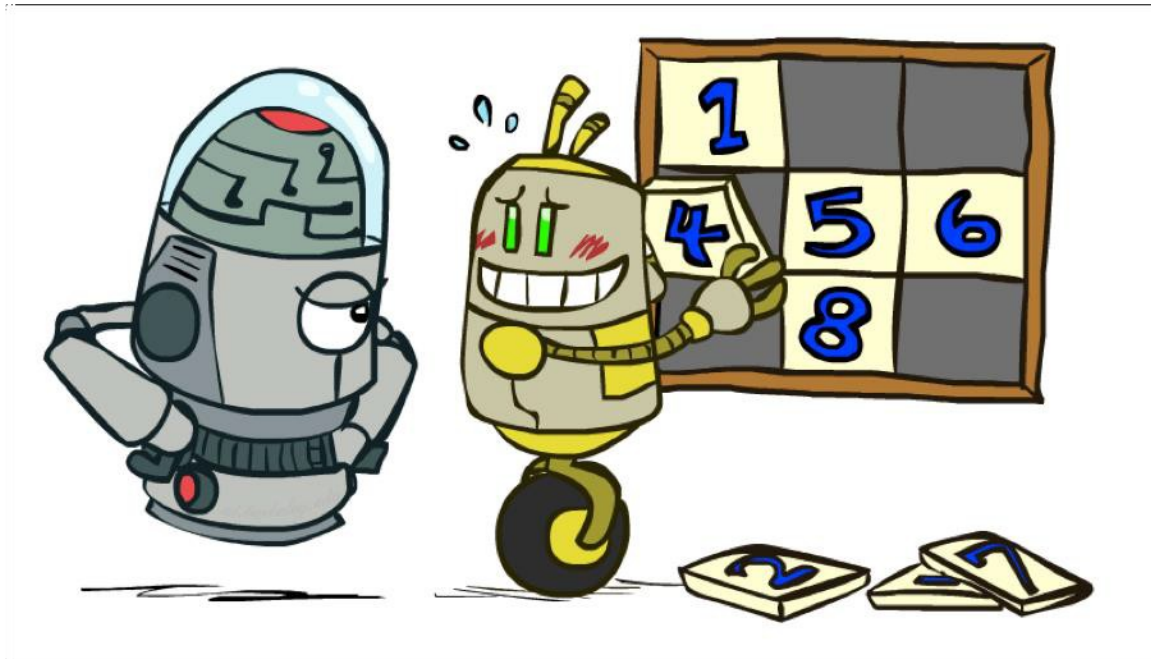
- Heuristic: $h_1(n) =$ 不在位的棋子数
- Why is it admissible?
- $h(\text{start}) = 8$



Start State



Goal State

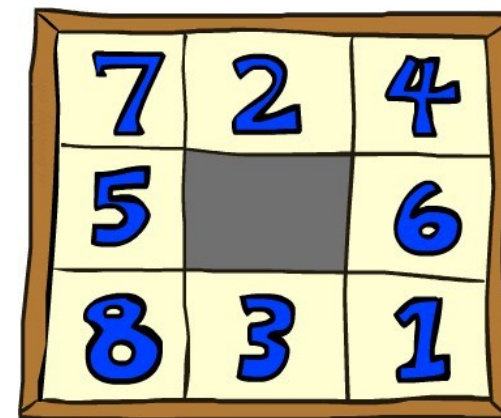


Average nodes expanded
when the optimal path has...

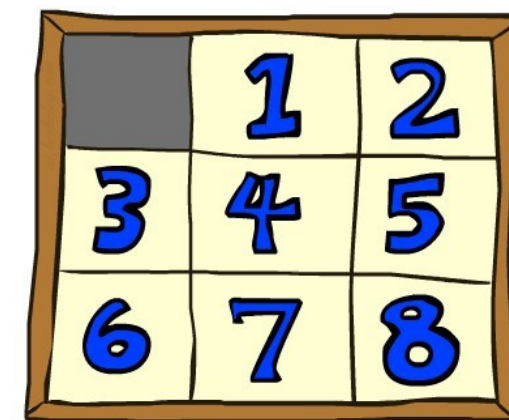
	...4 steps	...8 steps	...12 steps
UCS	112	6,300	3.6×10^6
TILES	13	39	227

8 Puzzle II

- Heuristic:
- $h_2(n)$ = 所有棋子到其目标位置的距离和



Start State



Goal State

- Why is it admissible?

□ $h(\text{start}) = 3 + 1 + 2 + \dots = 18$

Average nodes expanded when the optimal path has...			
	...4 steps	...8 steps	...12 steps
TILES	13	39	227
MANHATTAN	12	25	73

可采纳的启发式

E.g., 针对八数码问题，有两个常用的可采纳的启发式函数：

- $h_1(n)$ = 不在位的棋子数
- $h_2(n)$ = 所有棋子到其目标位置的距离和
-

7	2	4
5		6
8	3	1

Start State

	1	2
3	4	5
6	7	8

Goal State

- $h_1(\text{start}) = 8$
- $h_2(\text{start}) = 3+1+2+2+2+3+3+2 = 18$
- **$h^*(S) = 26$ (上述两类启发式函数都没有高估到达目标的实际代价)**

启发式函数性能对比

- 对于任意结点 n , 若 $h_2(n) \geq h_1(n)$, 称 h_2 比 h_1 占优势 , h_2 更优。

- 搜索代价 (扩展的平均结点数)

- $d=12$ IDS = 3,644,035 nodes

- $A^*(h_1) = 227$ nodes

- $A^*(h_2) = 73$ nodes

- $d=24$ IDS = too many nodes

- $A^*(h_1) = 39,135$ nodes

- $A^*(h_2) = 1,641$ nodes

d	搜索代价			有效分支因子		
	IDS	$A^*(h_1)$	$A^*(h_2)$	IDS	$A^*(h_1)$	$A^*(h_2)$
2	10	6	6	2.45	1.79	1.79
4	112	13	12	2.87	1.48	1.45
6	680	20	18	2.73	1.34	1.30
8	6384	39	25	2.80	1.33	1.24
10	47127	93	39	2.79	1.38	1.22
12	3644035	227	73	2.78	1.42	1.24
14	-	539	113	-	1.44	1.23
16	-	1301	211	-	1.45	1.25
18	-	3056	363	-	1.46	1.26
20	-	7276	676	-	1.47	1.27
22	-	18094	1219	-	1.48	1.28
24	-	39135	1641	-	1.48	1.26

设计接近又总是小于等于 $h^*(n)$ 的 $h(n)$ 是应用 A^* 算法 ($h(n) \leq h^*(n)$) 搜索问题解答的关键

A* 搜索的应用

- Video games
- Pathing / routing problems
- Resource planning problems
- Robot motion planning
- Language analysis
- Machine translation
- Speech recognition
- ...



总结：搜索策略

无信息搜索策略：只使用问题定义中提供的状态信息

- 宽度优先搜索
- 一致代价搜索
- 深度优先搜索
- 深度受限搜索
- 迭代加深搜索

有信息搜索策略：使用启发式信息指导搜索

- 贪婪最佳优先搜索
- A* 搜索