Heuristic Search Algorithm

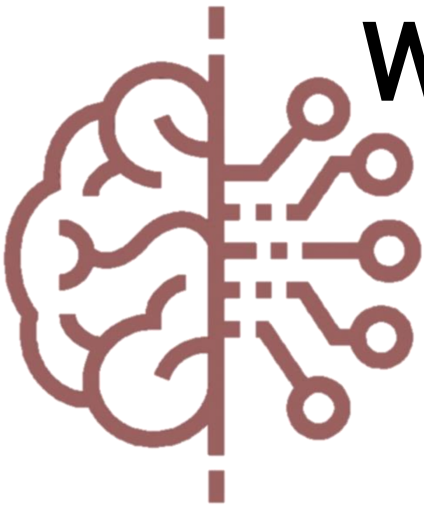
(Informed Search)

Artificial Intelligence

School of Computing
Universiti Teknologi Malaysia

Outline

1. What is Heuristic Search?
2. Selection of a Search Strategy
   - Hill climbing Search
   - Best-first Search
   - Search with Heuristic
   - Comparison of all search strategy
3. Heuristic Evaluation Function f(n)



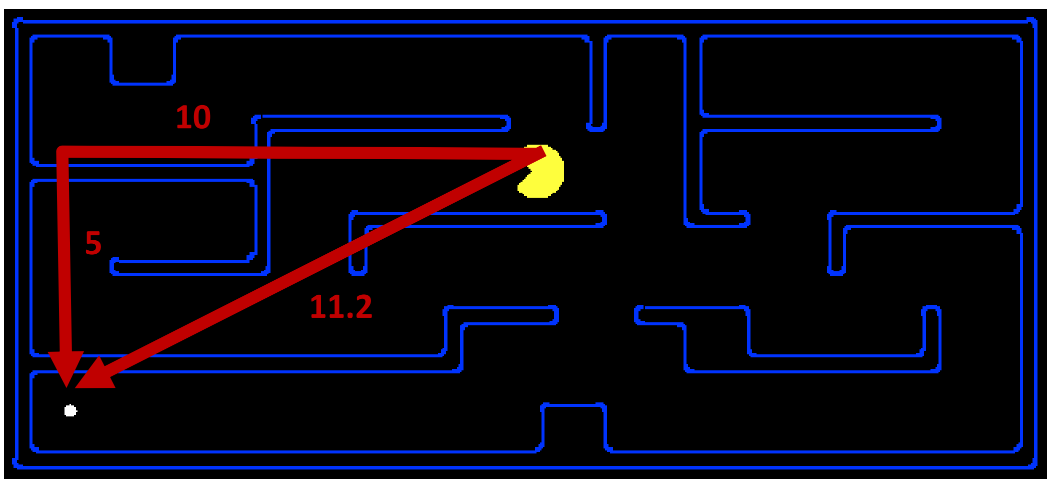
What is Heuristic Search?

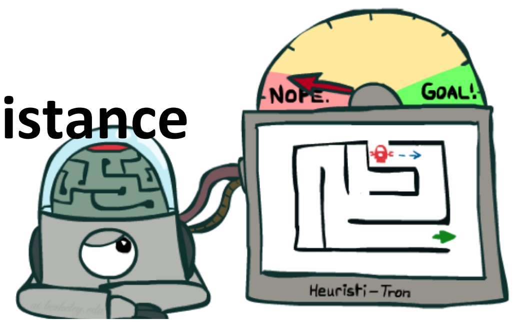
- **Definition**
- **Taxonomy of Heuristic Search**

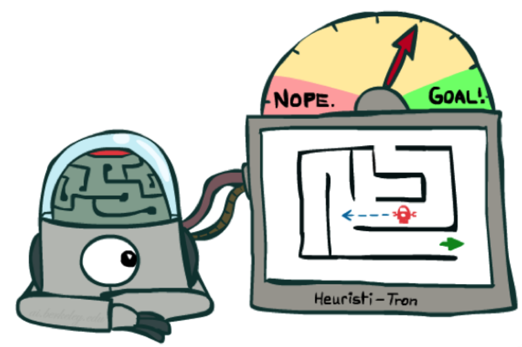
Heuristic Search: Definition

Heuristic – a technique designed to solve a problem quickly

- **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







Heuristic Search: Definition

Rule of thumb is a practical and approximate way of doing or measuring something

- A technique to solve a problem faster than classic methods, or to find an approximate solution when classic methods cannot.
- Heuristic is a rule of thumb that probably leads to a solution. Heuristics play a major role in search strategies because of exponential nature of the most problems. Heuristics help to reduce the number of alternatives from an exponential number to a polynomial number
- This is a kind of a shortcut as we often trade one of optimality, completeness, accuracy, or precision for speed.

Heuristic Search: Definition

- **(Rules of thumb): Weak search method because it is based on experience or intuition.**
- **Used to prune spaces of possible solution**
- **When to employ Heuristic?**

1. **A problem may not have an exact solution.**
   e.g. medical diagnosis: doctors use heuristic
2. **A problem may have an exact solution, but the computational cost of finding it may be prohibitive.**
   e.g in chess (exhaustive or brute force search)

Heuristic Search: Taxonomy

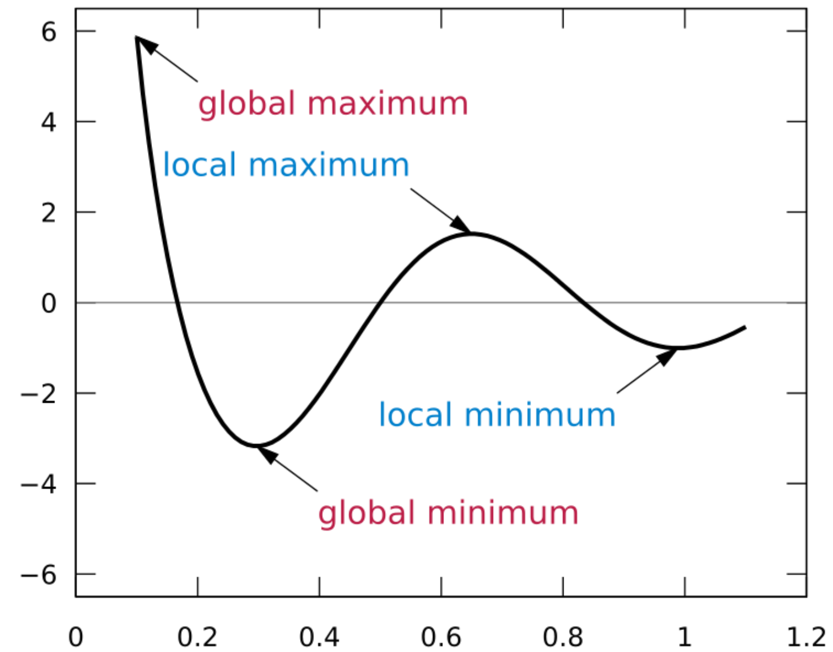
- Heuristic Search
  - Direct Techniques (Blind / Uninformed)
    - Breadth First Search (BFS)
    - Depth First Search (DFS)
  - Weak Techniques (Informed / Heuristic)
    - Best-First Search
    - A* Search
    - Bidirectional Search
    - Tabu Search
    - Beam Search
    - Simulated Annealing
    - Hill Climbing
    - Constraint Satisfaction Problems

Weak Heuristic Search

- **Hill climbing Search**
- **Best-first Search**
- **A* Search**

Hill Climbing

- A local search algorithm which continuously moves from solution to solution in the search space that offer better solution. It terminates when it reaches a peak value where no neighbour has a higher value.



Hill Climbing

- Simplest, the best child is selected for further expansion
- Limited memory, no backtracking and recovery
- Problem with hill climbing:
  - can lead along an infinite paths that fail.
  - Can stuck at local optima reach a state that is better evaluation than its children, the algorithm halts.
  - There is no guarantee optimal performance
- Advantage
  - Can be used effectively if the heuristic is sufficient

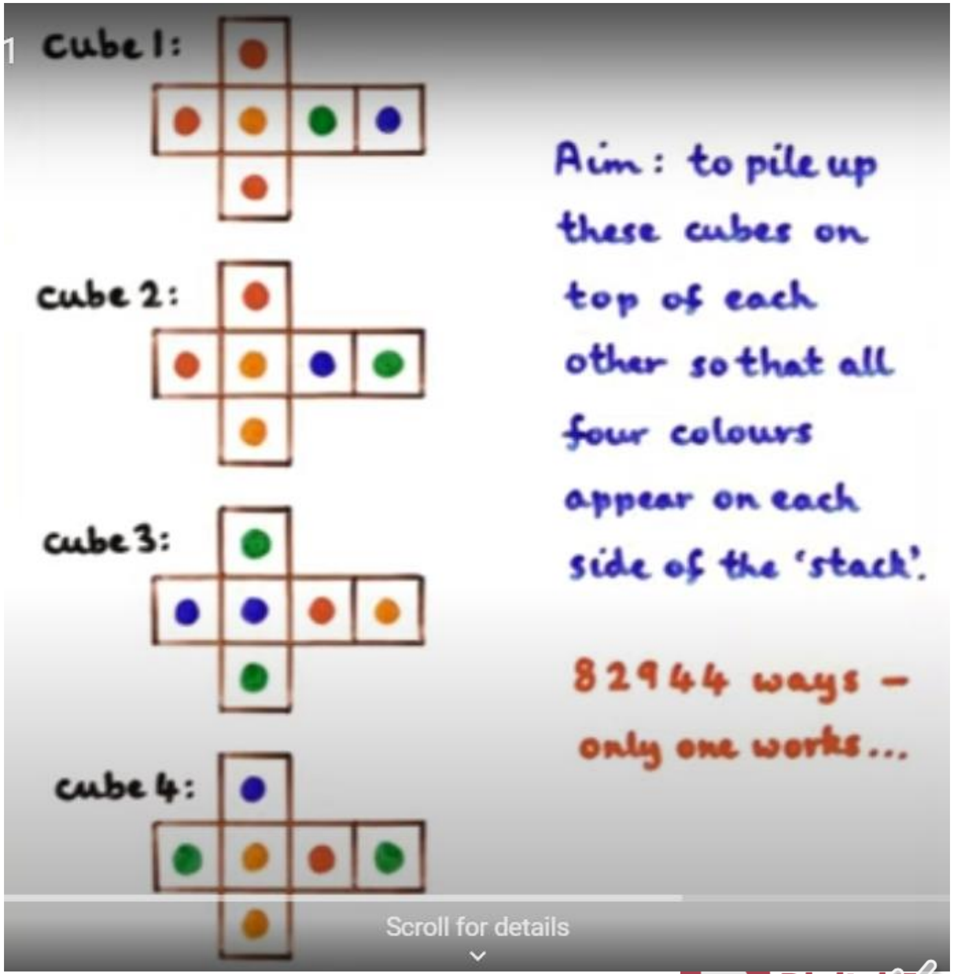
Hill Climbing

- Here the generate and test method is augmented by an heuristic function which measures the closeness of the current state to the goal state.
  1. Evaluate the initial state, if it is goal state, quit; otherwise current state is initial state.
  2. Select a new operator for this state and generate a new state.
  3. Evaluate the new state
     1. if it is closer to goal state than current state, make it current state
     2. if it is no better ignore
  4. If the current state is goal state or no new operators available, quit. Otherwise repeat from 2.

Consider the problem of four 6-sided cubes, and each side of the cube is painted in one of 4 colours. The 4 cubes are placed next to one another and the problem lies in arranging them so that the four available colours are displayed whichever way the 4 cubes are viewed.

Hill Climbing

- In the case of the four cubes a suitable heuristic is the sum of the no of different colors on each of the four sides, and the goal state is 16 four on each side. The set of rules is simply choose a cube and rotate the cube through 90 degrees. The starting arrangement can either be specified or is at random.



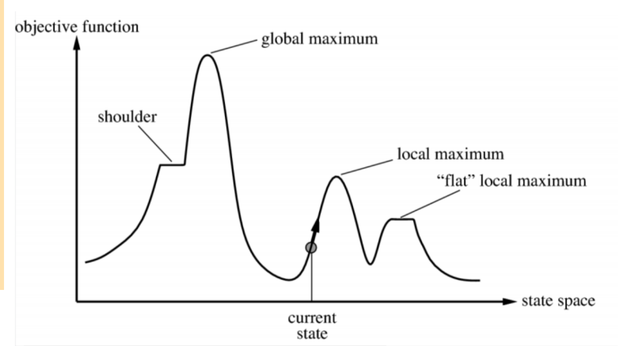
Hill Climbing: Type

Simple Hill climbing	• It examines the neighboring nodes one by one and selects the first neighboring node which optimizes the current cost as next node.
Steepest-Ascent Hill climbing	• It first examines all the neighboring nodes and then selects the node closest to the solution state as of next node.
Stochastic hill climbing	• It does not examine all the neighboring nodes before deciding which node to select. It just selects a neighboring node at random and decides (based on the amount of improvement in that neighbor) whether to move to that neighbor or to examine another.

UTM
UNIVERSITI TEKNOLOGI MALAYSIA

Hill Climbing: Issues

www.utm.my



Local Optima

- All neighboring states have values worse than the current. The greedy approach means we won't be moving to a worse state. This terminates the process even though there may have been a better solution. As a workaround, we use backtracking.

Plateau

- All neighbors to it have the same value. This makes it impossible to choose a direction. To avoid this, we randomly make a big jump.

Ridge

- At a ridge, movement in all possible directions is downward. This makes it look like a peak and terminates the process. To avoid this, we may use two or more rules before testing.

Premier Digital Tech University™

Best First Search

- A combination of depth first and breadth first searches.
- DFS - a solution can be found without computing all nodes and BFS - it does not get trapped in dead ends.
- The best first search allows us to switch between paths thus gaining the benefit of both approaches. At each step the most promising node is chosen. If one of the nodes chosen generates nodes that are less promising it is possible to choose another at the same level and in effect the search changes from depth to breadth.
- If on analysis these are no better then this previously unexpanded node and branch is not forgotten and the search method reverts to the descendants of the first choice and proceeds, backtracking as it were.

Best First Search

- It is a general algorithm for heuristically searching any state space graph
- Supports a variety of heuristic evaluation functions
- Better and flexible algorithm for heuristic search
- Avoid local optima, dead ends; has open and close lists
- selects the most promising state
- apply heuristic and sort the ‘best’ next state in front of the list (priority queue) can jump to any level of the state space
- If lead to incorrect path, it may retrieve the next best state

UTM
UNIVERSITI TEKNOLOGI MALAYSIA

Best First Search

www.utm.my

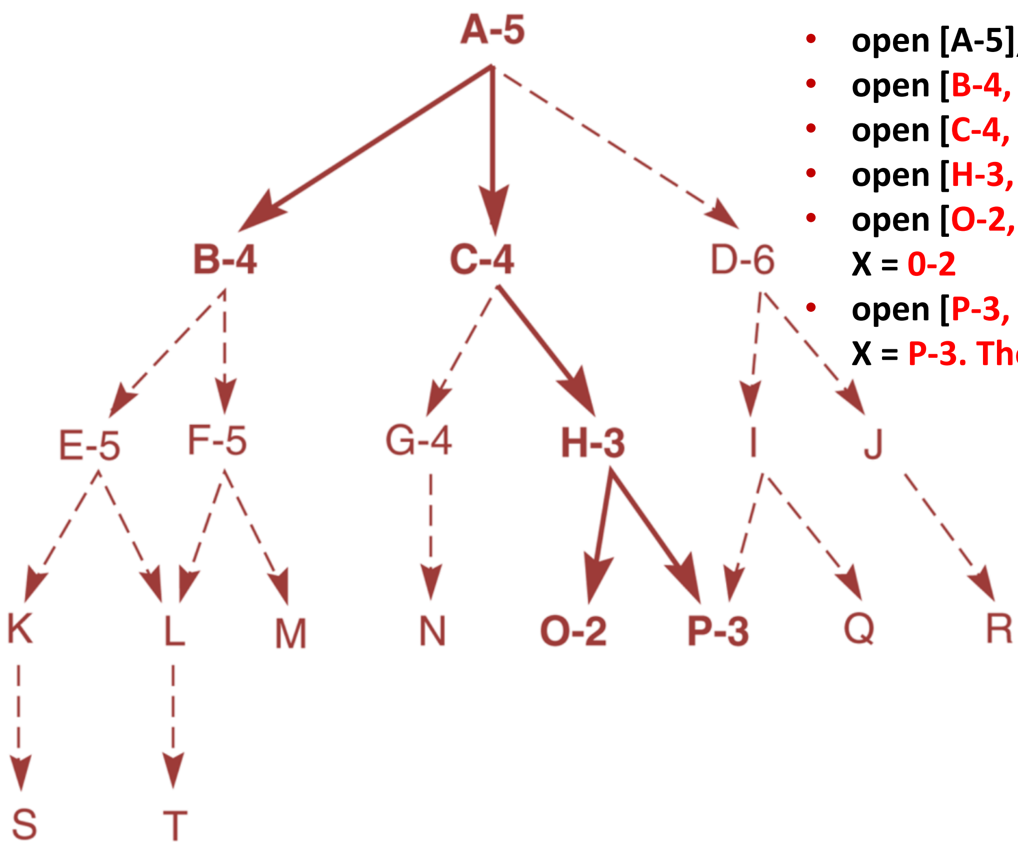
```
function best_first_search;

begin
   open := [Start];                                                 % initialize
   closed := [ ];
   while open ≠ [ ] do                                          % states remain
      begin
         remove the leftmost state from open, call it X;
         if X = goal then return the path from Start to X
         else begin
               generate children of X;
               for each child of X do
               case
                     the child is not on open or closed:
                        begin
                              assign the child a heuristic value;
                              add the child to open
                        end;
                     the child is already on open:
                        if the child was reached by a shorter path
                        then give the state on open the shorter path
                     the child is already on closed:
                        if the child was reached by a shorter path then
                           begin
                              remove the state from closed;
                              add the child to open
                           end;
               end;                                                       % case
               put X on closed;
               re-order states on open by heuristic merit (best leftmost)
            end;
return FAIL                                                    % open is empty
end.
```

Premier Digital Tech University™

innovative • entrepreneurial • global

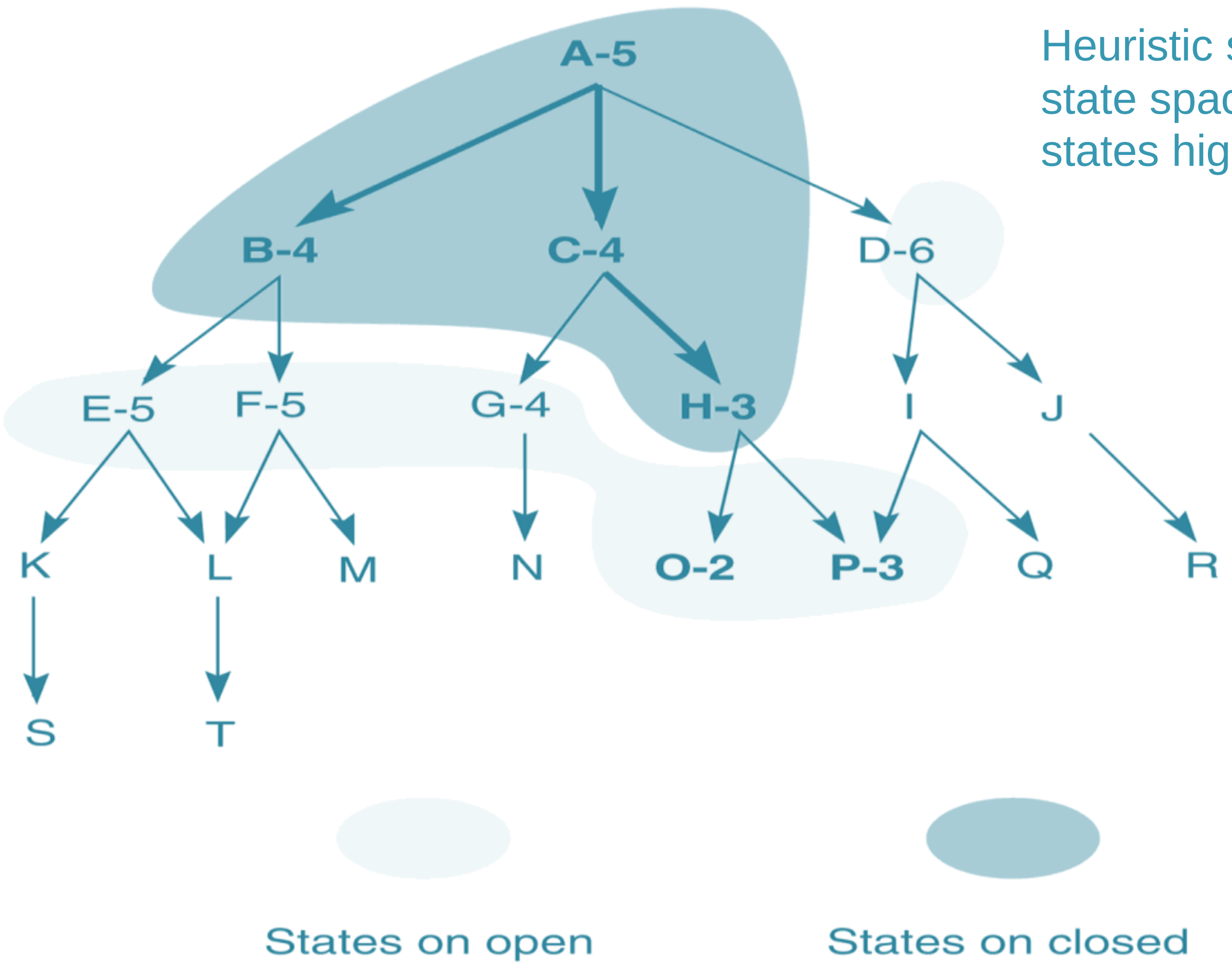
Best First Search

Example: Trace the following search space with the BFS algorithm. The numbers associated with the state names give the **heuristic value** of the state. **Goal = P-3**



- **open [A-5], closed = [ ], X = A-5**
- **open [B-4, C-4, D-6], closed = [A-5], X = B-4**
- **open [C-4, E-5, F-5, D-6], closed = [B-4, A-5], X = C-4**
- **open [H-3, G-4, E-5, F-5, D-6], closed = [C-4, B-4, A-5], X = H-3**
- **open [O-2, P-3, G-4, E-5, F-5, D-6], closed = [H-3, C-4, B-4, A-5], X = 0-2**
- **open [P-3, G-4, E-5, F-5, D-6], closed = [O-2, H-3, C-4, B-4, A-5], X = P-3. The solution is found**

Best First Search



Heuristic search of a hypothetical state space with open and closed states highlighted.

A* Search

What is A* Search Algorithm?

- **A* Search algorithm is one of the best and popular technique used in path-finding and graph traversals.**

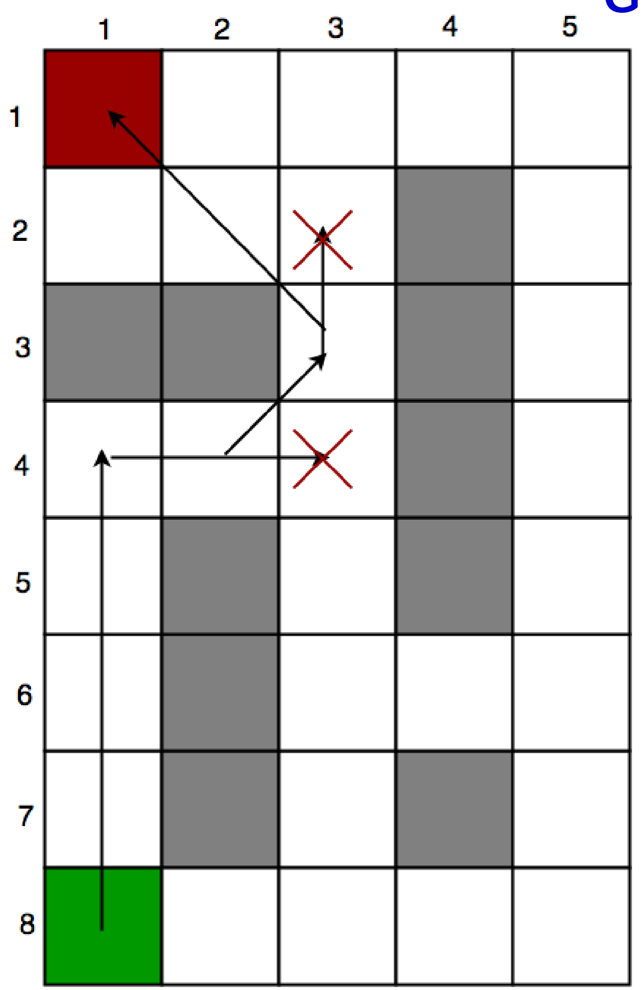
Why A* Search Algorithm?

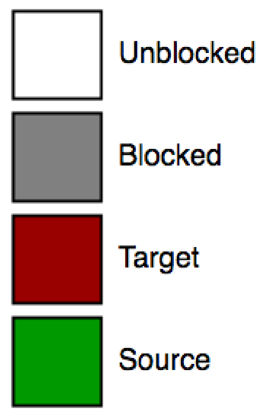
- **Unlike other traversal techniques, it has “brains” - a smart algorithm which separates it from the other conventional algorithms.**
- **Many games and web-based maps use this algorithm to find the shortest path very efficiently (approximation).**

A* Search

A square grid with obstacles

Goal: to reach the target cell as quickly as possible.





A* Search Algorithm makes the most intelligent choice at each step. Hence you can see that algorithm goes from (4,2) to (3,3) and not (4,3) (shown by cross).

Similarly the algorithm goes from (3,3) to (2,2) and not (2,3) (shown by cross).

Approximation Heuristics

There are generally three approximation heuristics to calculate h

1. **Manhattan Distance**
2. **Diagonal Distance**
3. **Euclidean Distance**

Heuristic Evaluation Function f(n)

Heuristic Evaluation Function f(n)

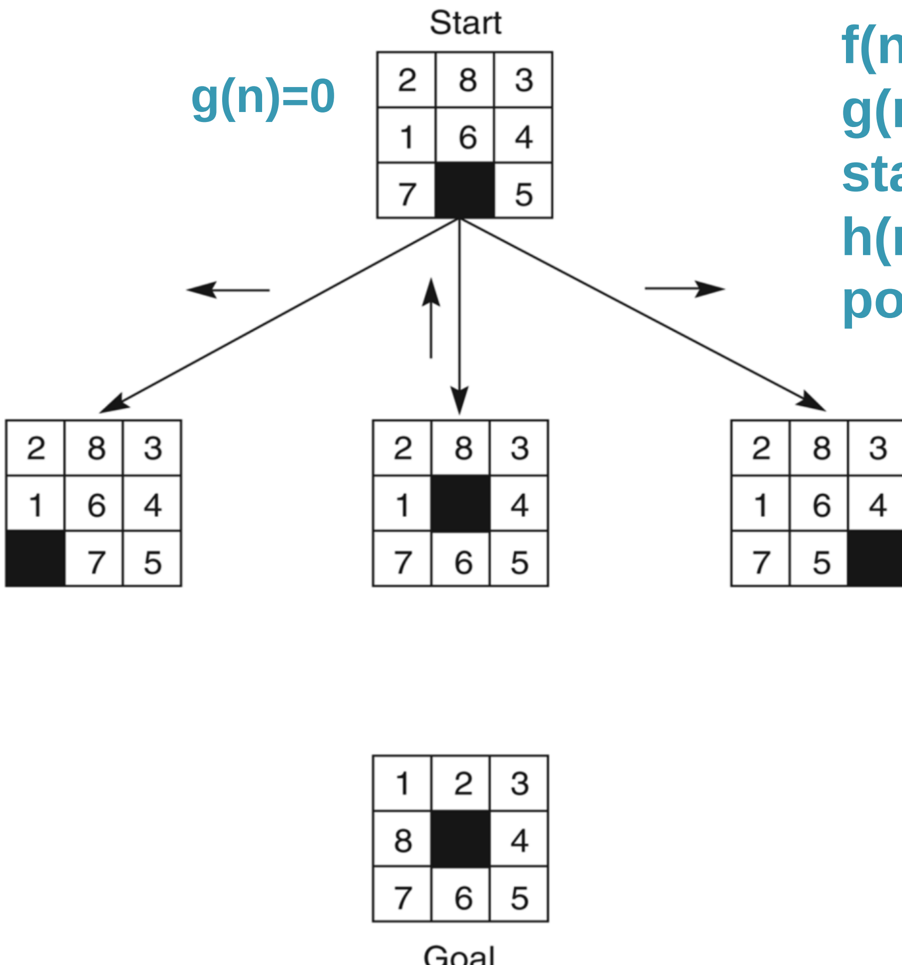
- **To evaluate performances of heuristics for solving a problem**
- **Goal - good heuristic using limited information to make intelligent choices**
- **To better heuristic, $f(n) = g(n) + h(n)$;**
  ***g*(*n*) is distance from start to *n*,**
  ***h*(*n*) is distance from *n* to goal**
- **Eg. 8 puzzle, heuristics h(n) could be:**
  - **No. of tiles in wrong position**
  - **No. of tiles in correct position**
  - **Number of direct reversal (2X)** (2 tiles must be swapped to be in order)
  - **Sum of distances out of place**
- **And g(n) is the depth measure**

Heuristic Evaluation Function f(n)

The start state, first set of moves, and goal state for an 8-puzzle instance.

g(n)=0

g(n)=1



f(n) = g(n) + h(n)

g(n) = actual dist. from n to start

h(n) = no. of tiles in wrong position

Activity: Heuristic Evaluation Function f(n)

Three heuristics applied to states in the 8 puzzle

- Devising good heuristics is sometimes difficult; OUR GOAL is to use the limited information available to make INTELLIGENT CHOICE

	State	Tiles out of place	Sum of distances out of place	2 x the number of direct tile reversals
g(n) = ?	2 8 3 1 6 4 7 5	?	?	0
g(n) = ?	2 8 3 1 4 7 6 5	?	?	0
g(n) = ?	2 8 3 1 6 4 7 5	?	?	0

h(n)

Use formula as follows:
$f(n) = g(n) + h(n)$
g(n) = actual dist. from n to start
h(n) = no. of tiles in wrong position

1	2	3
8		4
7	6	5

Goal

Premier Digital Tech University™

Activity: Open & Closed states

- In the tree of 8 puzzle given in the next slide, Give the value of f(n) for each state, based on g(n) and h(n)

f(n) = g(n) + h(n)
g(n) = actual dist. from n to start
h(n) = no. of tiles in wrong position

- Trace using best first search, what will be the lists of open and closed states? Complete the lists below until successfully achieving the goal.

1. **open = [a4];**
   closed = [ ]
2. **open = [c4, b6, d6];**
   closed = [a4]
3. **open = [e5, f5, b6, d6, g6];**
   closed = [a4, c4]