Introduction to Artificial Intelligence

Unit # 4

Acknowledgement

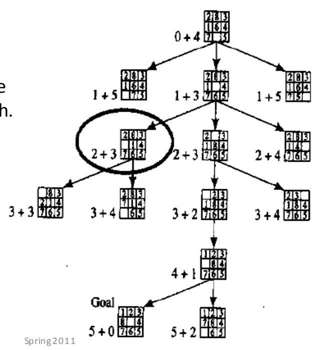
- The slides of this lecture have been taken from the lecture slides of CS307 – "Introduction to Artificial Intelligence" by Dr. Sajjad Haider.

A* search

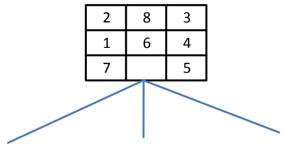
- Idea: avoid expanding paths that are already expensive
- Evaluation function *f(n) = g(n) + h(n)*
  - *g(n)* = cost so far to reach *n*
  - *h(n)* = estimated cost from *n* to goal
  - *f(n)* = estimated total cost of path through *n* to goal

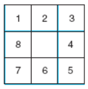
A* Search (Using Heuristic # 1)

- We add a "depth factor" to f: f(n) = g(n) + h(n).
- g(n) is an estimate of the "depth" of n in the graph.
- h(n) is a heuristic evaluation of node n.



A* Search
(Using Heuristic # 2)



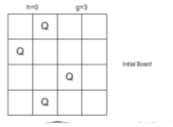


Goal State

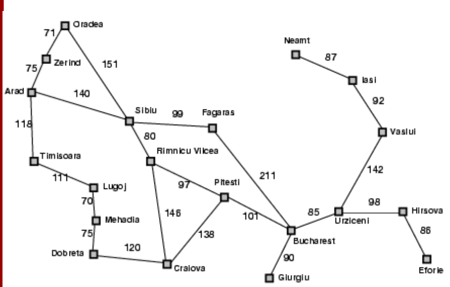
- Expand the tree

N-Queen

- The N-queens problem is a search problem where the desired result is an N by N board with N queens such that no queen threatens another
- The problem can be simplified by assigning a queen to each row on the board.
- Enumerating the search space is then defined as looking at the possible moves of queens horizontally.



Romania with step costs in km
(Russell & Norvig)

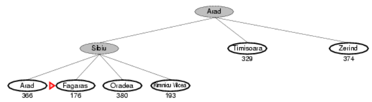


Straight–line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	10
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

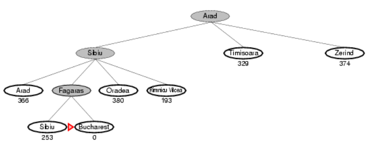
Best-first search example

Best-first search example

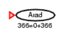
Best-first search example



Best-first search example

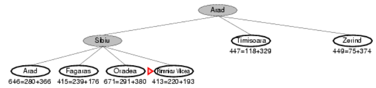


A* search example

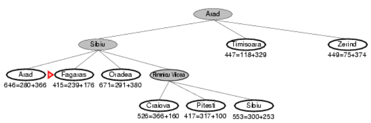


A* search example

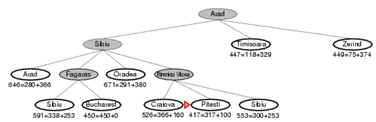
A* search example



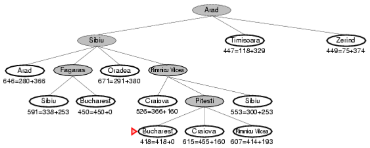
A* search example



A* search example

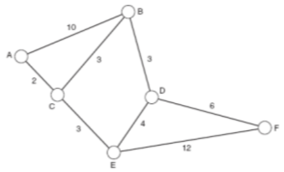


A^* search example



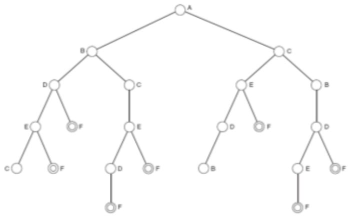
The Modified Traveling Salesman Problem

- Each node represents a town, and the vertices between nodes represent roads that join towns together.
- A is the starting node, and F is the goal node.
- Each vertex is labeled with a distance, which shows how long that road is.
- The aim of this problem is to find the shortest possible path from city A to city F.



Search Tree for the Modified TSP

1 A,B,D,E,F
2 A,B,D,F
3 A,B,C,E,D,F
4 A,B,C,E,F
5 A,C,E,F
6 A,C,B,D,E,F
7 A,C,B,D,F

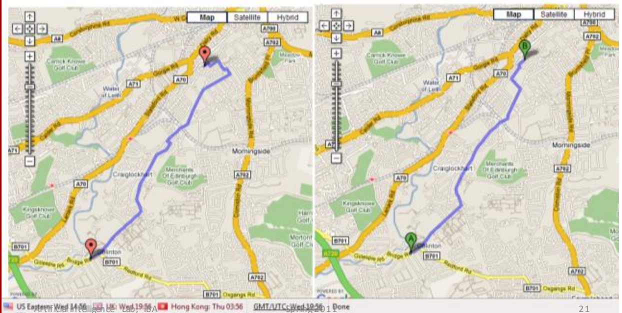


Search Demo

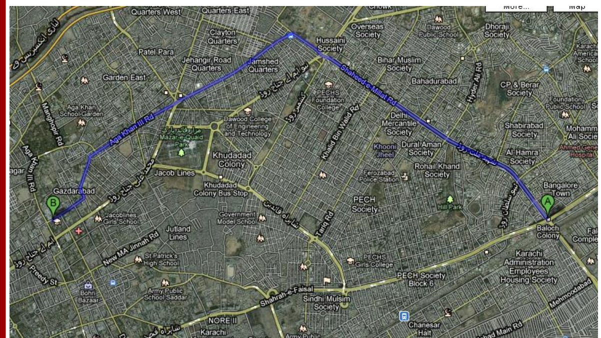
http://www.cs.rochester.edu/u/kautz/Mazes/search_algorithm_demo.htm

Result

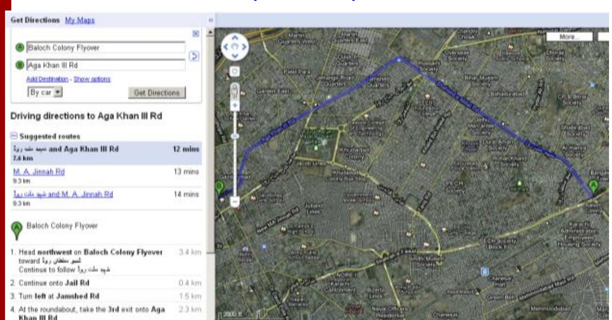
A star agorithm **Google Map API**



Karachi Path Finding by Google



Karachi Path Finding by Google (Cont'd)



Karachi Path Finder Developed at IBA

Informed vs. Uniformed Search

- A search method or heuristic is **informed** if it uses additional information about nodes that have not yet been explored to decide which nodes to examine next.
- If a method is not informed, it is **uninformed** or **blind**.
- Best-first search is an example of informed search, whereas breadth-first and depth-first search are uninformed or blind.
- The more informed a search method is, the more efficiently it will search.

Summary: Search

- Depth-first search and breadth-first search are extremely commonly used and well understood exhaustive search methods.
- A search method is **complete** if it will always find a solution if one exists. A search method is **optimal** (or admissible) if it always finds the best solution that exists.
- Heuristics can be used to make search methods more informed about the problem they are solving. A heuristic is a method that provides a better guess about the correct choice to make at any junction that would be achieved by random guessing.
- One heuristic is more informed than another heuristic if a search method that uses it needs to examine fewer nodes to reach a goal.

More Search……☺
Greedy Algorithm

(Source: Wikipedia)

- Greedy algorithms can be characterized as being 'short sighted', and as 'non-recoverable'. They are ideal only for problems which have 'optimal substructure'. Despite this, greedy algorithms are best suited for simple problems (e.g. giving change).
- **Greedy choice property**
- We can make whatever choice seems best at the moment and then solve the subproblems that arise later. The choice made by a greedy algorithm may depend on choices made so far but not on future choices or all the solutions to the subproblem. It iteratively makes one greedy choice after another, reducing each given problem into a smaller one. In other words, a greedy algorithm never reconsiders its choices.

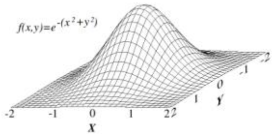
Hill Climbing

(Source: Wikipedia)

- In computer science, **hill climbing** is a mathematical optimization technique which belongs to the family of local search. It is an iterative algorithm that starts with an arbitrary solution to a problem, then attempts to find a better solution by incrementally changing a single element of the solution. If the change produces a better solution, an incremental change is made to the new solution, repeating until no further improvements can be found.
- For example, hill climbing can be applied to the traveling salesman problem. It is easy to find an initial solution that visits all the cities but will be very poor compared to the optimal solution. The algorithm starts with such a solution and makes small improvements to it, such as switching the order in which two cities are visited. Eventually, a much shorter route is likely to be obtained.
- Hill climbing is good for finding a local optimum (a good solution that lies relatively near the initial solution) but it is not guaranteed to find the best possible solution (the global optimum) out of all possible solutions (the search space).

Local vs Global Optimum?

What is the optimum point?



Local vs Global Optimum?

What do you think about the optimum point now?