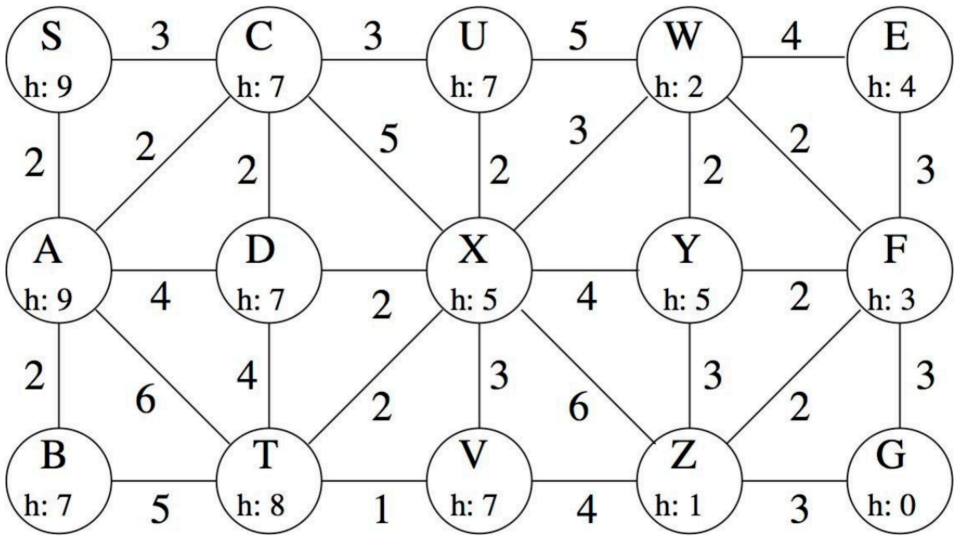
Week 2 : Heuristic Path Search (Open leering Week 4)

Tutorial: Heuristic Path SearchSince Heuristic Path Search has already been covered in lectures, you are invited to proceed to Activities 4.1, 4.2, 4.3, 4.4, 4.5 from the Week 4

Tutorial 4: Heuristic Path Search

4.1: Path Search Algorithms on a Graph (Activity 4.1)

Consider the task of finding a path from start state S to goal state G, given the distances and heuristic values in this diagram:



For each of the following strategies, list the order in which the states are **expanded.** Whenever there is a choice of states, you should select the one that comes first in alphabetical order. In each case, you should skip any states that have previously been expanded, and you should continue the search until the goal node is expanded.

1. Breadth First Search
2. Depth First Search
3. Uniform Cost Search [Hint: first compute for each state in the graph]
4. Greedy Search, using the heuristic shown
5. A*Search, using the heuristic shown

4.2 A* Search for the 8-Puzzle (Activity 4.2):

Complete the A*Search for the 8-Puzzle activity on the Finding Heuristics page. Discuss your findings with the group.

4.3: Relationships Between Search Strategies (Activity 4.3)

1. (Ex 3.22, R&N) Prove each of the following statements, or give a counterexample:
2. Breadth First Search is a special case of Uniform Cost Search.
3. Breadth First Search, Depth First Search and Uniform Cost Search are special cases of best-first search.
4. Uniform Cost Search is a special case of A*Search.

4.4: Heuristic Path Algorithm (Activity 4.4)

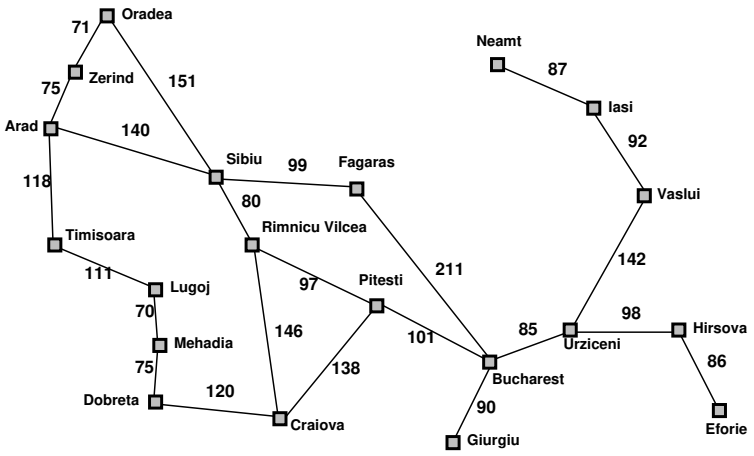
(Ex 3.28, R&N) The heuristic path algorithm is a best-first search in which the objective function is:

$$f(n) = (2-w)\,g(n) + w\,h(n), \quad \text{where } 0 \le w \le 2.$$

What kind of search does this perform when $w = 0$? when $w = 1$? when $w = 2$?
For what values of w is this algorithm complete? For what values of w is it optimal, assuming $h()$ is admissible?

4.5: This exercise concerns the route-finding problem using the Romania map from Russell & Norvig (Artificial Intelligence: A Modern Approach) as an example.

Define the route-finding problem (from Arad to Bucharest) as a state space search problem (give short descriptions of the state space, etc. in English). What order are nodes in the state space expanded for each of the following algorithms when searching for a (shortest) path between Arad and Bucharest (when there is a choice of nodes, take the one earliest in the alphabetical ordering)? Make sure you understand the key properties of the different algorithms, as listed below.



Straight−line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

To clarify, for breadth-first search, stop the search when the goal state is generated and use a check to ensure that nodes with the same state as a previously expanded node are not added to the frontier. For the other search algorithms, stop the search when the goal state is expanded; for uniform-cost search include a check that nodes with the same state as a previously expanded nodes are not added to the frontier (as in breadth-first search) and a test

so that only one node for a given state is stored on the frontier (that with the shortest path to that state), and for depth-first search and its variants use cycle checking along a path to avoid repeated states that may lead to infinite branches.

(i) Depth-first search (efficient use of space but may not terminate)
(ii) Breadth-first search (space inefficient, guaranteed to find a solution)
(iii) Uniform-cost search (similar to breadth-first, but order nodes by cost)
(iv) Iterative deepening depth-first search (space efficient, but repeated work)
(v) Greedy best-first search (efficient, not guaranteed optimal solution)
(vi) A^* search with straight-line distance heuristic (inefficient, guaranteed optimal solution)

Which algorithm is suitable in practice for solving route-finding problems such as this?

Students activity

Activity: 4.5 Understanding Informed Search Algorithms with Mazes

Discuss your findings and insights from the 'Fun with Mazes' activity. Compare your findings and discuss any discrepancies.

(a) an environment for which Bidirectional Search would find a solution faster than Breadth First Search.

(b) an environment for which Greedy Search takes much longer than A*Search.

(c) an environment for which Greedy Search produces a path that is much longer than the optimal path.

(d) an environment for which A*Search with the Euclidean Distance heuristic takes much longer than with the Manhattan Distance heuristic.

(e) an environment that is interesting for some other reason.