Tutorial for week 5

1. We saw in lectures a graph representing the road map of part of Romania; the cost of a path we take to be the distance via the road as given on the graph. We also have a table of straight-line distances from town to Bucharest.

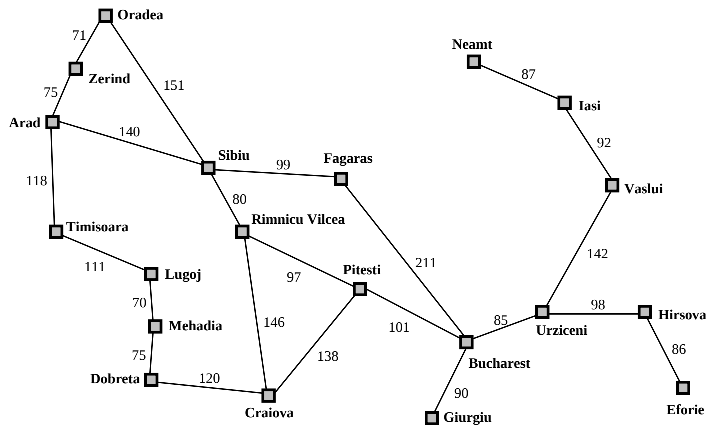

Figure 1: Some Romanian roads

Arad	366	**Mehadia**	241
Bucharest	0	**Neamt**	234
Craiova	160	**Oradea**	380
Dobreta	242	**Pitesti**	100
Eforie	161	**Rimnicu Vilcea**	193
Fagaras	176	**Sibiu**	253
Giurgiu	77	**Timisoara**	329
Hirsova	151	**Urziceni**	80
Iasi	226	**Vaslui**	199
Lugoj	244	**Zerind**	374

Figure 2: h_{SLD}: straight-line distance to Bucharest

(a) Show that using greedy best-first search with the heuristic function h_{SLD} does *not* give an optimal solution when looking for a path from Arad to Bucharest (this example was used in the lecture).

(b) Use greedy best-first search to look for a path from Iasi to Fagaras; (you should estimate the straight-line distance from the "map" above). This should reveal two problems with this strategy; what are they?

(c) We can use A^* search in this problem; h_{SLD} is an admissible heuristic that can be combined with the actual distance of the path so far g to get a new heuristic f. Show that f finds an optimal solution in part (a), and solves one of the problems in part (b).

2. A heuristic $h(n)$ is said to be *consistent* if and only if for every node n and every successor n' generated by an action a, the estimated cost of reaching the goal from n is not more than the actual cost of getting from n to n' plus the estimated cost of getting from n' to the goal That is:

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

where $c(n, a, n')$ is the cost associated with action a that moves from state n to state n'.

(a) Explain why the straight-line heuristic used above is consistent.

(b) Show that every consistent heuristic is admissible, i.e. it always underestimates the true cost of reaching the goal.

One way to do this is to consider a sequence of actions leading from a given node n_0 to goal node n_k:

$$n_0 \xrightarrow{a_0} n_1 \xrightarrow{a_1} n_2 \ \ldots \ n_{k-1} \xrightarrow{a_{k-1}} n_k.$$

This gives a way of checking that a heuristic is admissible and so that A^* search with such a heuristic is optimal.

3. We saw that the Manhattan distance of tiles from their target positions gives an admissible heuristic. Invent a heuristic for the 8-puzzle that sometimes overestimates, and show how it can lead to a suboptimal solution on a given problem, using A^* search. (You need to provide start and target positions, and show that the solution found is sub-optimal. The heuristic should always return non-negative values, and return 0 for the goal state.)