CS/CNS/EE 154: Articificial Intelligence
Problem Set 1

Handed out: 10 Oct 2010
Due: 22 Oct 2010

---

1 A^* search [10 points]

Trace the operation of A^* search applied to the problem of getting to Bucharest from Lugoj using the straight line distance heuristic. That is, show the sequence of nodes that the algorithm will consider and the f, g and h score for each node.

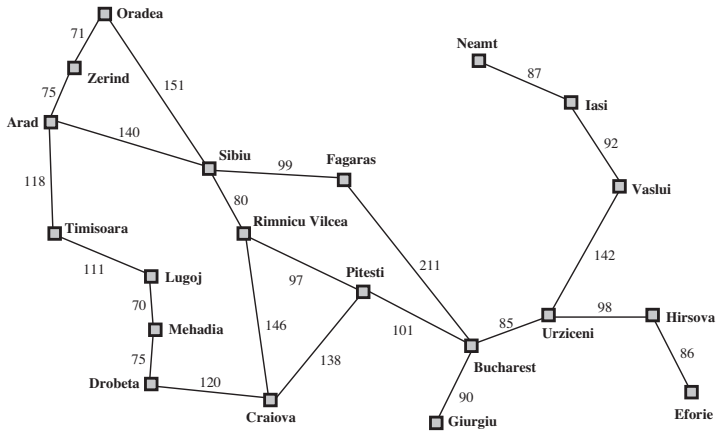

Figure 1: A simplified road map of part of Romania.

2 Rendevouz in Romania [25 points]

Suppose two friends live in different cities on a map, such as the Romania map shown in Figure 1. On every turn, we can simultaneously move each friend to a neighboring city on the map. The amount of time needed to move from city i to neighbor j is equal to the road distance $d(i, j)$ between the cities, but on each turn the friend that arrives first must wait until the other one arrives (and calls the first on his/her cell phone) before the next turn can begin. We want the two friends meet as quickly as possible.

a. Formalize this search problem, i.e., specify the environment states, the actions, the successor functions for each action, and the performance measure. You will find it helpful to define some formal notation here.

b. Let $D(i, j)$ be the straight-line distance between cities i and j. Which of the following heuristic functions are admissible? (i) $D(i, j)$; (ii) $2 \cdot D(i, j)$; (iii) $D(i, j)/2$.

c. Are there completely connected maps for which no solution exists?

d. Are there maps in which all solutions require one friend to visit one city twice?

3 Ghost vs. Pac-Man [35 points]

Now suppose that in Problem 2, instead of trying to find each other, one of the friends (for convenience called "Pac-Man") tries to escape from the other friend (for convenience called "Ghost"). The problem then becomes a two-player **pursuit-evasion** game. Instead of moving simultaneously, we assume now that the players take turns moving. The game ends only when the players are on the same node; the terminal payoff to the pursuer is minus the total time taken. (The evader "wins" by never losing.) In the following, the map that indicates the initial position of the pursuer (P) and the evader (E), as well as the possible moves, is given in Figure 2 (a).

a. Copy (or print) the (partial) game tree Figure 2 (b), and mark the values of the terminal nodes.

b. Next to each internal node, write the strongest fact you can infer about its value (a number, one or more inequalities such as "≥ 14", or a "?")

c. Beneath each question mark, write the name of the node reached by that branch.

d. Explain how a bound on the value of the nodes in (c) can be derived from consideration of shortest-path lengths on the map, and derive such bounds for these nodes. Remember the cost to get to each leaf as well as the cost to solve it.

e. Now suppose that the tree as given, with the leaf bounds from (d), is evaluated from left to right. Circle those "?" nodes that would *not* need to be expanded further, given the bounds from part (d), and cross out those that need not be considered at all.

f. Can you prove anything in general about who wins the game on a map that is a (finite) tree?

g. Are there finite graphs in which the evader is always able to escape, as long as pursuer is not able to catch the evader in her first move?

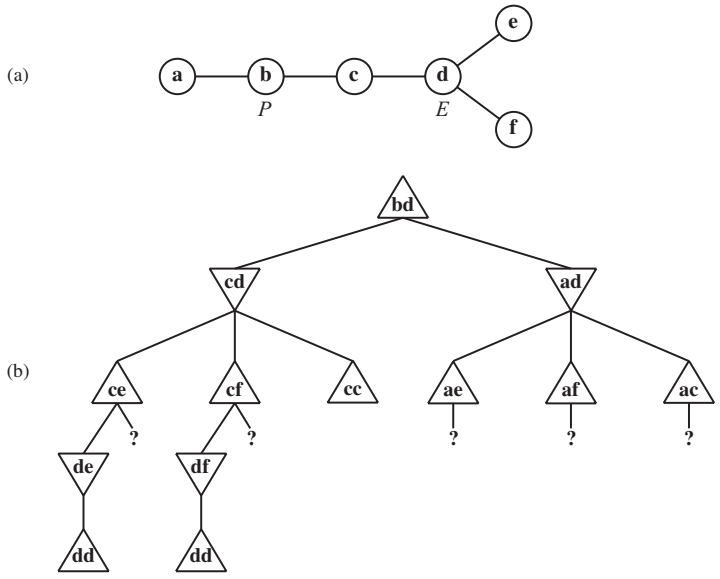

Figure 2: (a) A map where the cost of every edge is 1. Initially the pursuer P is at node **b** and evader E is at node **d**. (b) A partial game tree for this map. Each node is labeled with the P, E positions. P moves first. Branches marked with "?" have yet to be explored.

4 Propositional Logic and Resolution [30 points]

A propositional 2-CNF expression is a conjunction of clauses, each containing *exactly* 2 literals, e.g.,

$$(A \vee B) \wedge (\neg A \vee C) \wedge (\neg B \vee D) \wedge (\neg C \vee G) \wedge (\neg D \vee G)$$

a. Prove using resolution that the above sentence entails G.

b. Two sentences are *semantically distinct* if they are not logically equivalent. How many semantically distinct 2-CNF clauses can be constructed from n proposition symbols?

c. Using your answer to (b), prove that propositional resolution always terminates in time polynomial in n given a 2-CNF sentence containing no more than n distinct symbols.

d. Explain why your argument in (c) does not apply to 3-CNF.