Informatics 2D: Tutorial 2

Adversarial Search and Informed Search*

Week 3

1 Adversarial Search

This exercise was taken from R&N Chapter 5.

Consider the two-player game shown in Figure 1.

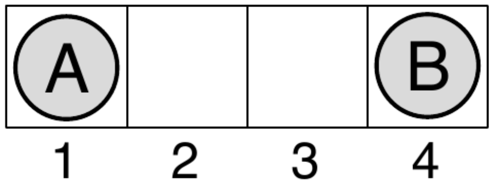


Figure 1: The starting position of a simple game. Player A moves first. The two players take turns moving, and each player must move their token to an open adjacent space in either direction. If the opponent occupies an adjacent space, then the player may jump over the opponent to the next available space. For example, if A is on 3 and B is on 2, then A may move back to 1. The game ends when one player reaches the opposite end of the board. If player A reaches space 4 first, then the value of the game to A is $+1$; if player B reaches space 1 first the value of the game to A is -1.

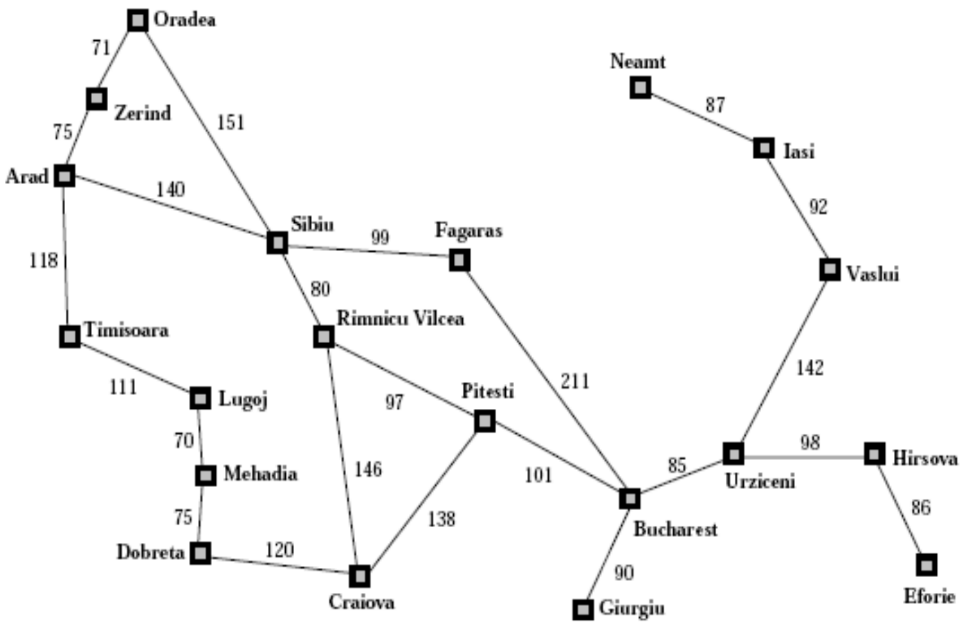
1. Draw the complete game tree, using the following conventions:
   - Write each state as (S_A, S_B) where S_A and S_B denote token locations
   - Put each terminal state in square boxes and write its game value in a circle
   - Put *loop states* (states that already appear on the path to the root) in double square boxes. Since it is not clear how to assign values to loop states, annotate each with a "?" in a circle.
2. Now mark each node with its backed-up minimax value (also in a circle). You will have to think of a way to assign values to the loop states.

*Credits: Kobby Nuamah, Michael Rovatsos

3. Explain why the standard minimax algorithm would fail on this game tree and briefly sketch how you might fix it, drawing on your answer to item (2) above. Does your modified algorithm give optimal decisions for all games with loops?

2 Informed Search

We saw in the lectures a graph representing the road map of part of Romania, see Figure 2. The cost of a path is the distance via the road, as given on the graph. We also have a table of straight-line distances from each town to Bucharest.



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: The Romania map from R&N with a table of straight-line distances to Bucharest.

a. Show that using greedy best-first search with the straight-line heuristic function, h_{SLD}, does not give an optimal solution when looking for a path from Arad to Bucharest.

b. Suppose that you have the following straight-line distances from Fagaras to: Neamt 140km, Iasi 175km, Vaslui 175km, Urziceni 180km, Hirsova 230km, Giurgiu 220km, Pitesti 50km, Rimnicu Vilcea 50km, Craiova 180km, Sibiu 60km. What happens when you try to use greedy best-first search to find a path from Iasi to Fagaras?

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 to get a new heuristic f. Show that f finds an optimal solution in part (a) and solves the problem in part (b).

3 Heuristics

(Taken from R&N Chapter 3)

Sometimes there is no good evaluation function for a problem, but there is a good comparison method: a way to tell whether one node is better than another, without assigning numerical values to either. Show that this is enough to do a Best-First search. Is there an analogue for A^* ?

4 *More to learn[1]

a. What are the differences among A^*, greedy best-first search, and Dijkstra's algorithm?

b. Assume, an agent is to travel across a square grid from the left-top corner to the right-bottom corner. The agent can move left, right, up, down only. In terms of the number of steps, the path via the left-bottom corner (one turn) and a path that is zigzagging downwards near the diagonal of the square ($2n-3$ turns) are equivalent. How can you modify the A^* algorithm such that a path with a smaller number of directional changes is preferred?

c. Not a recent paper, but still interesting: D. S. Nau (1983) Pathology on game trees revisited, and an alternative to minimaxing. *Artificial Intelligence* **21**(1-2), 221–244.

[1]Starred *problems are outside the examinable course content. Feel free to ignore them completely.