Artificial Intelligence — Solutions
Exercise Set 1

10p 1. Read the following statements and decide whether they are TRUE or FALSE.

a. The Turing test evaluates a system's ability to act rationally. False
b. It is possible for an agent to act rationally in an environment that is only partially observable. True
c. An agent's environment is said to be stochastic if the next state is completely determined by the current state and the agent's action. False
d. In a zero-sum two player game there is necessarily always a winner. False
e. Breadth-first search where all arcs have a cost of one will always find the shortest path to a goal if one exists. True
f. For a search problem, the path returned by uniform cost search may change if we add a positive constant c to every step cost. True
g. Depth-first search is an optimal, uninformed search technique. False
h. A greedy, best-first search algorithm is always complete. False
i. Using the minimax procedure with and without alpha-beta pruning will always identify the best move for the player whose turn it is to move. True
j. The alpha-beta algorithm is preferred to minimax because it computes the same answer as minimax while usually doing so without examining as much of the game tree. True

5p 2. Prove each of the following statements, or give a counterexample:

1p a. Breadth-first search is a special case of uniform-cost search.

If all step costs are the same, then g(n) d.p. depth(n), which means uniform-cost search emulates breadth-first search.

3p b. Breadth-first search, depth-first search, and uniform-cost search are special cases of best-first search.

Breadth-first search is best-first search with f(n) = depth(n).
Depth-first search is best-first search with f(n) = −depth(n).
Uniform-cost search is best-first search with f(n) = g(n).

1p c. Uniform-cost search is a special case of A^* search.

Uniform-cost search is A^* search with h(n) =0.

Arad	366	**Mehadia**	241
Bucharest	0	**Neamt**	234
Craiova	160	**Oradea**	380
Drobeta	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

Table1. Straight-line distances to Bucharest.

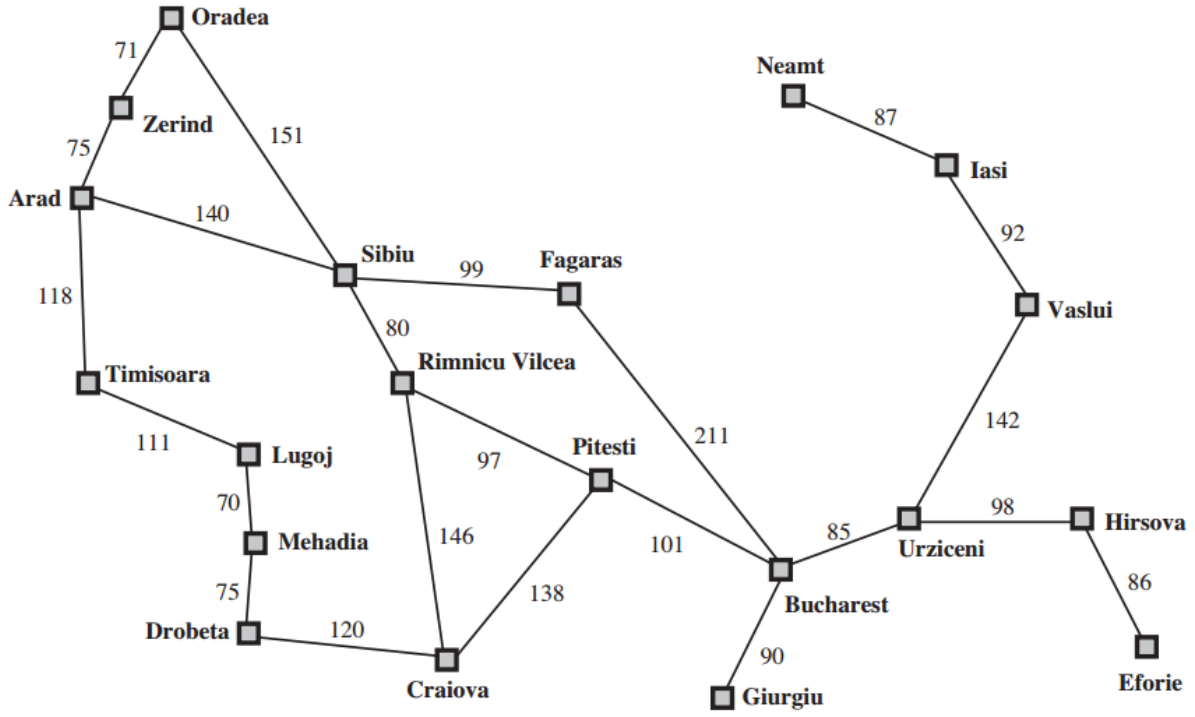
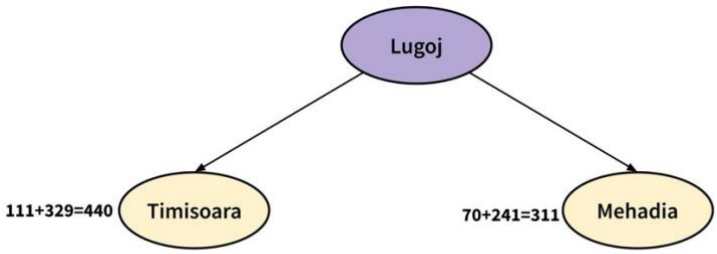


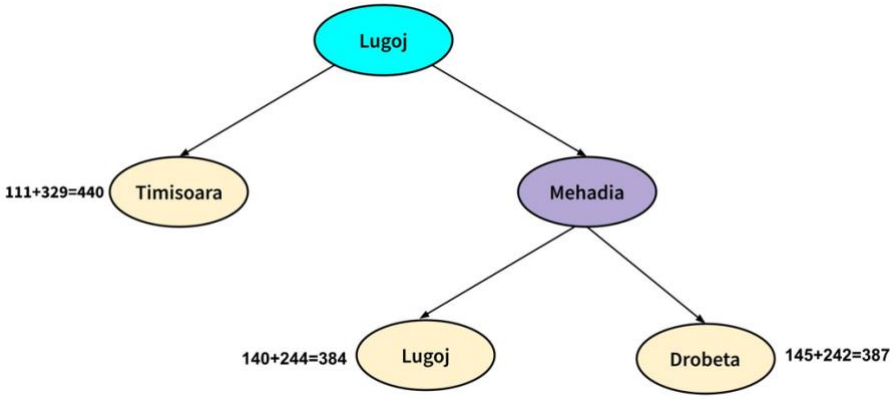
Fig.1. A simplified road map of part of Romania.

5p 3. Given the simplified road map of part of Romania in Fig.1 above, trace the operation of A∗ search, applied to the problem of reaching Bucharest from Lugoj, using the straight-line distance heuristic, in Table 1. Show the sequence of nodes that the algorithm will consider and the f, g, and h score for each node. For the sake of more easily managing the task, you might want to build a table.

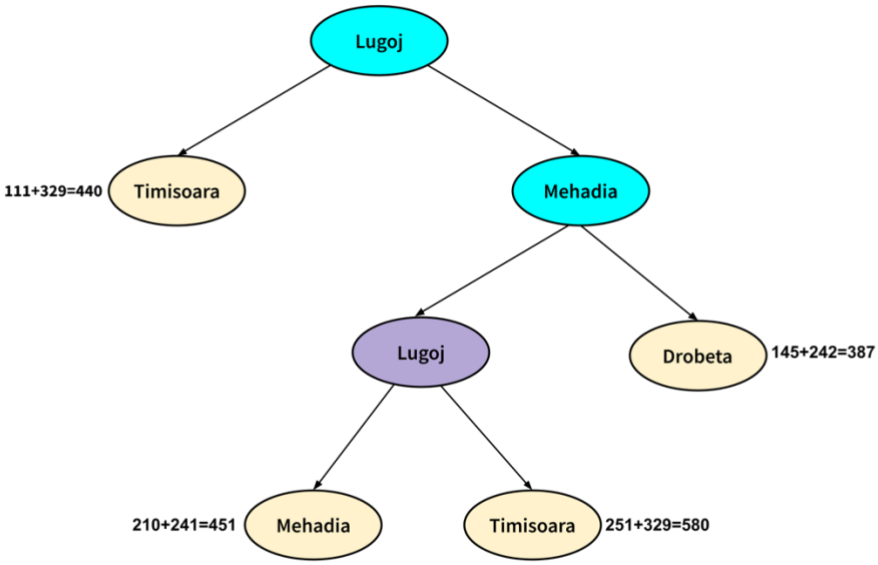
L[0+244=244]
M[70+241=311], T[111+329=440]



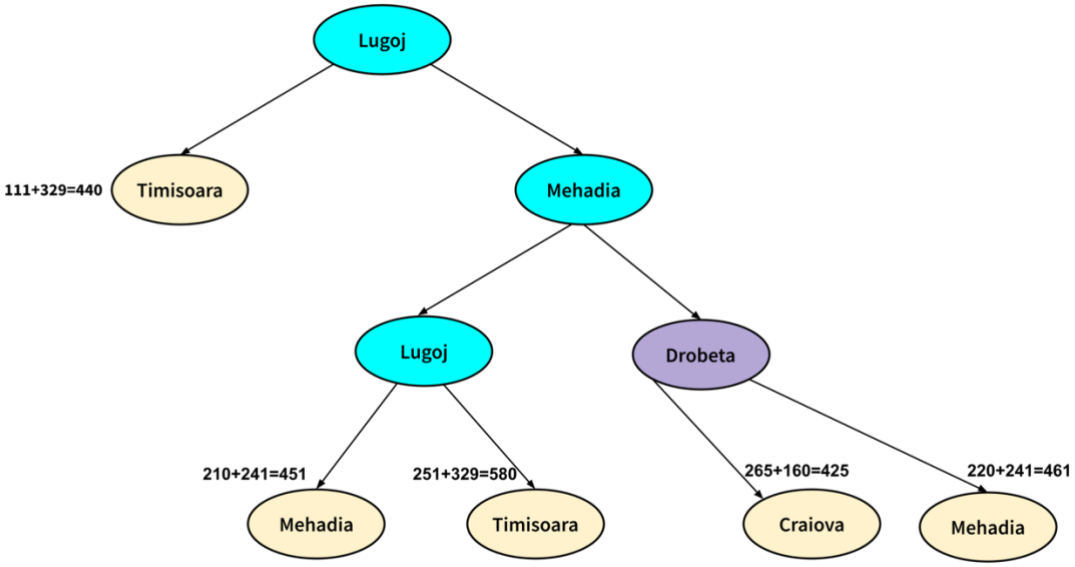
L[140+244=384], D[145+242=387], T[111+329=440]



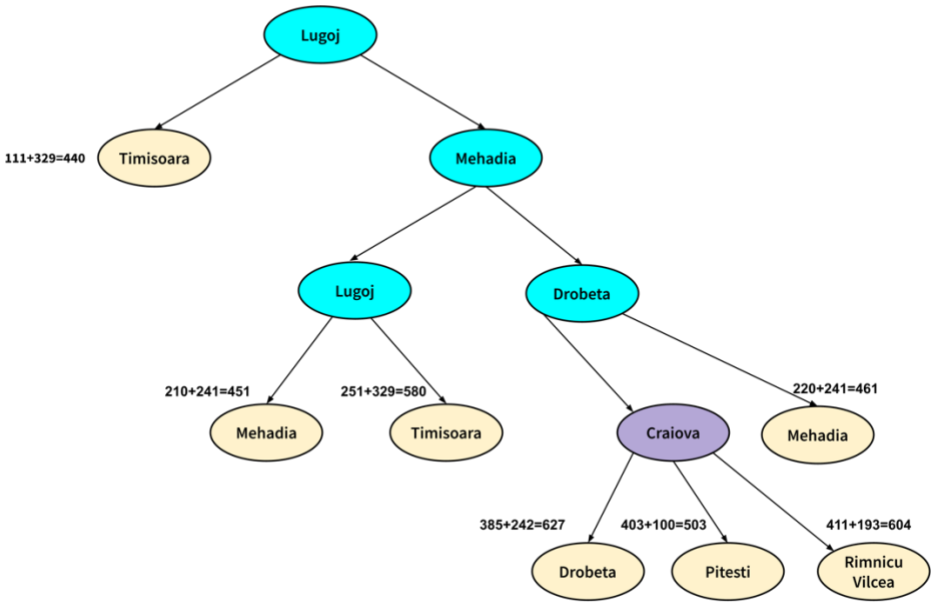
D[145+242=387], T[111+329=440], M[210+241=451], T[251+329=580]



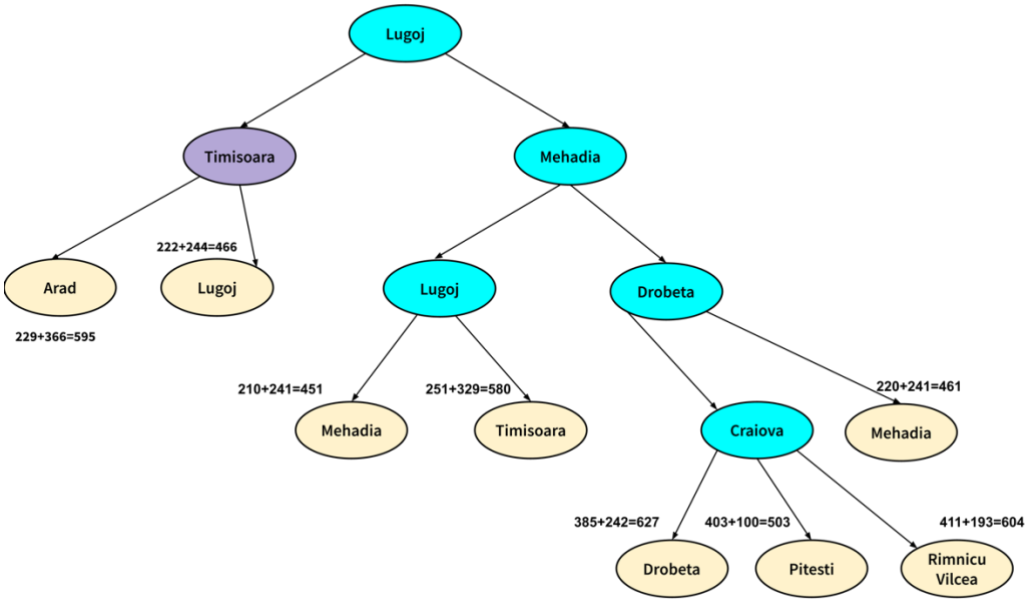
C[265+160=425], T[111+329=440], M[210+241=451], M[220+241=461], T[251+329=580]



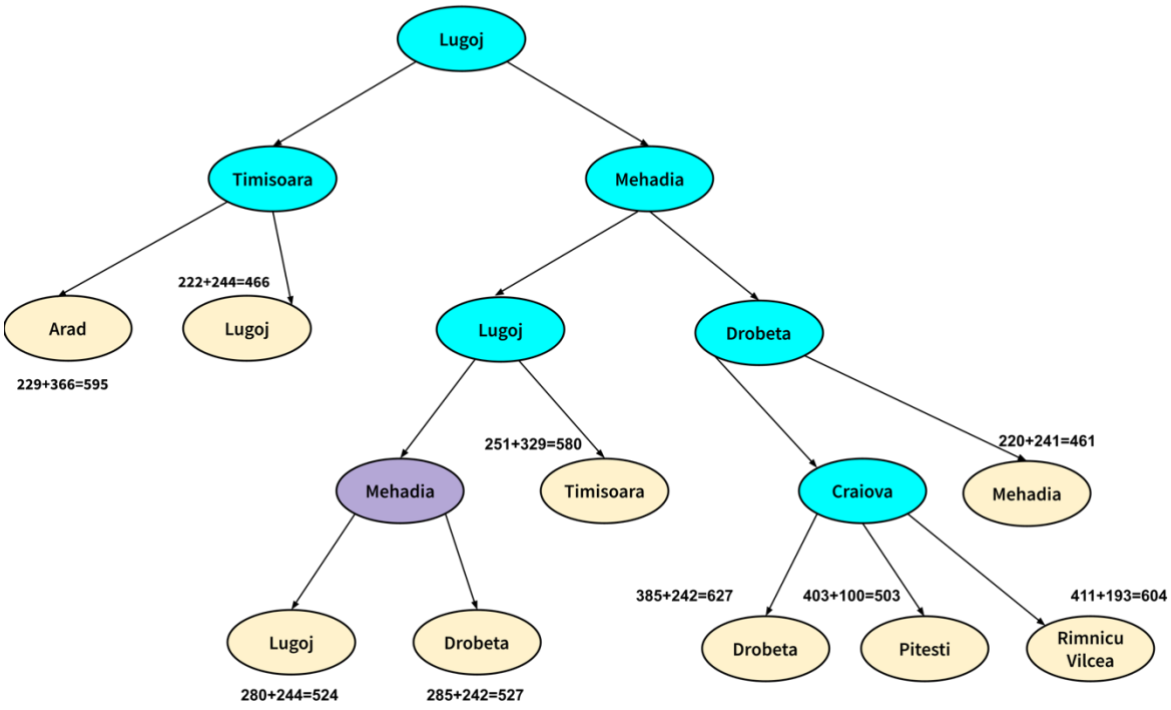
T[111+329=440], M[210+241=451], M[220+241=461], P[403+100=503], T[251+329=580], R[411+193=604], D[385+242=627]



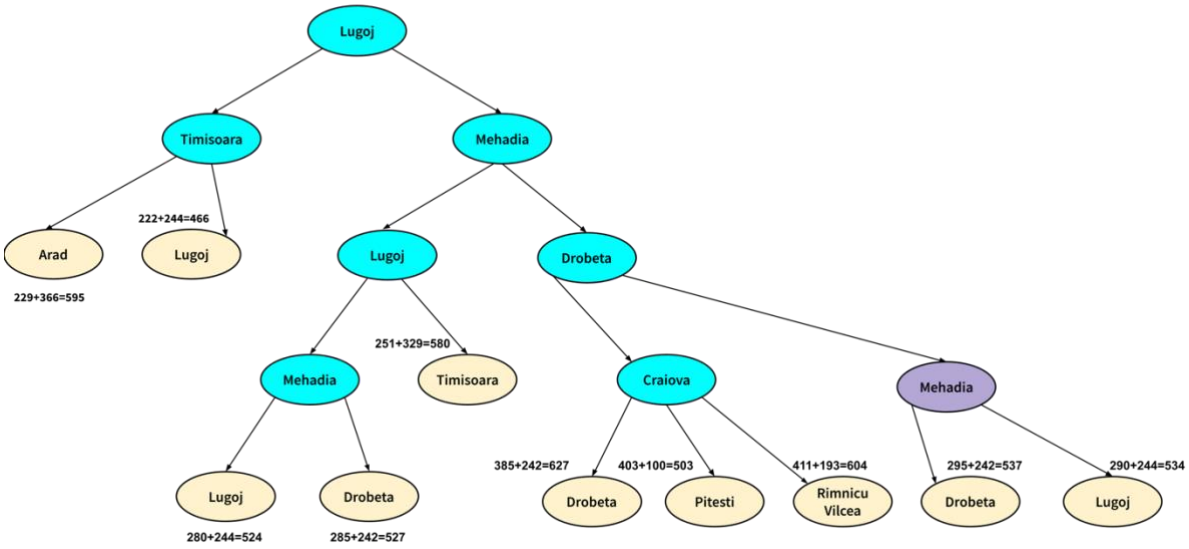
M[210+241=451], M[220+241=461], L[222+244=466], P[403+100=503], T[251+329=580], A[229+366=595], R[411+193=604], D[385+242=627]



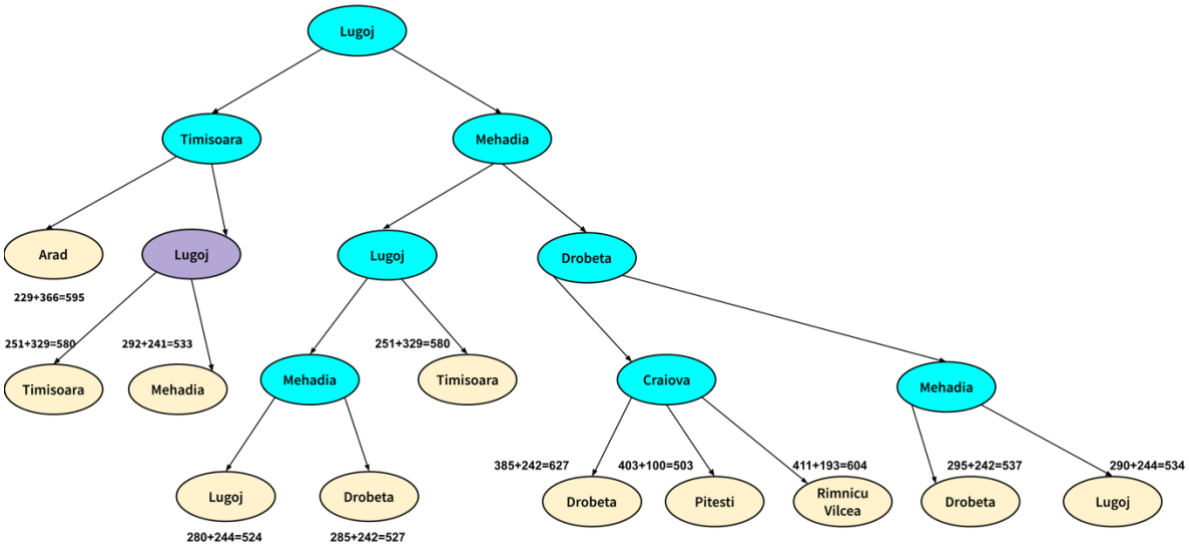
M[220+241=461], L[222+244=466], P[403+100=503], L[280+244=524], D[285+242=527], T[251+329=580], A[229+366=595], R[411+193=604], D[385+242=627]



L[222+244=466], P[403+100=503], L[280+244=524], D[285+242=527], L[290+244=534], D[295+242=537], T[251+329=580], A[229+366=595], R[411+193=604], D[385+242=627]

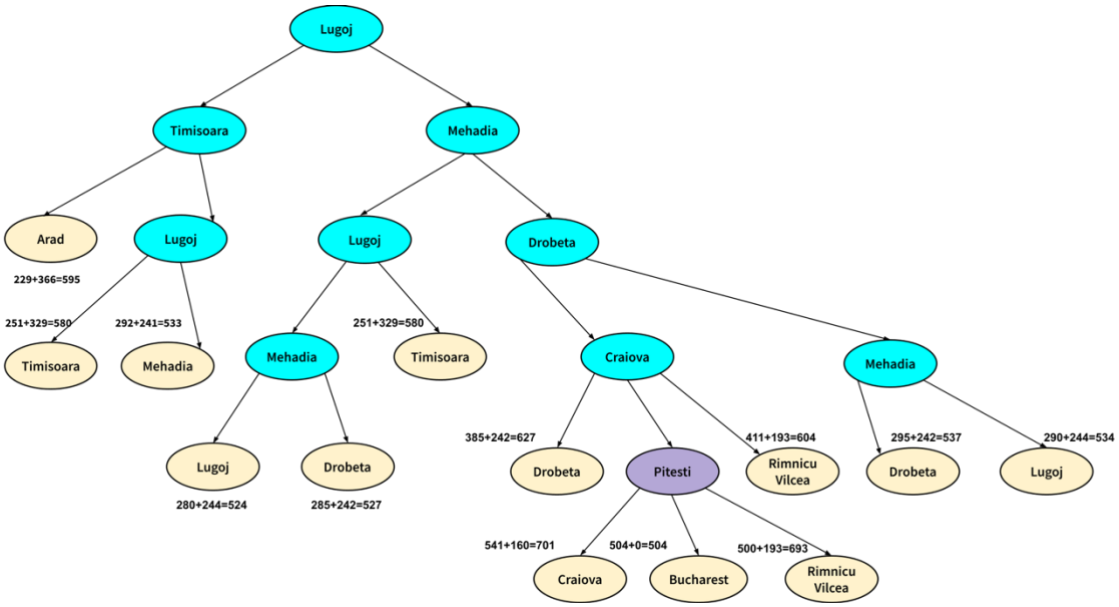


P[403+100=503], L[280+244=524], D[285+242=527], M[292+241=533], L[290+244=534], D[295+242=537],
T[251+329=580], A[229+366=595], R[411+193=604], D[385+242=627], T[333+329=662]

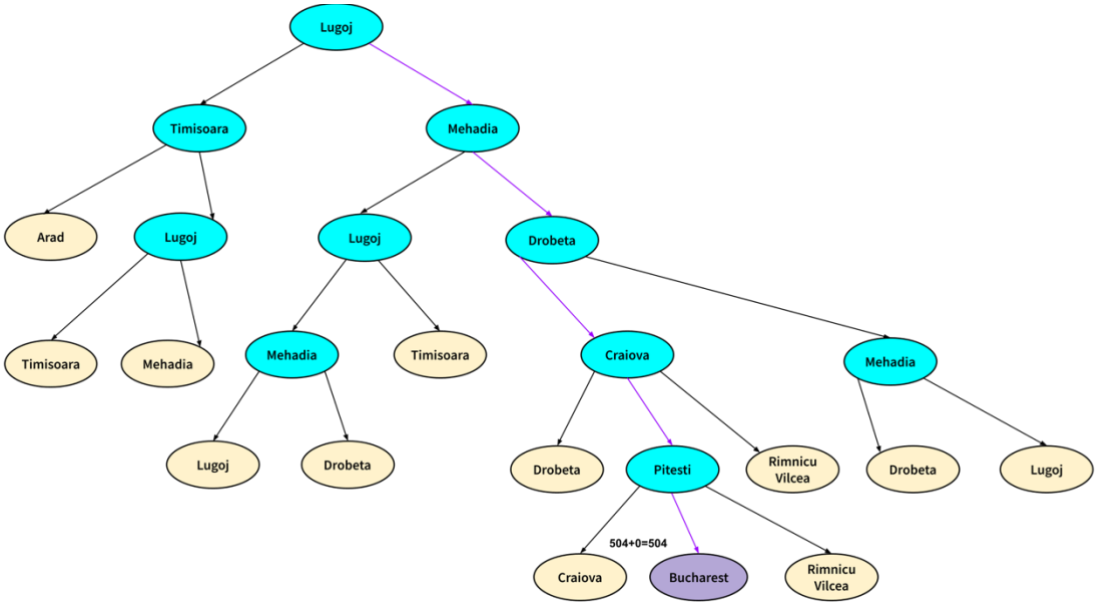


B[504+0=504], L[280+244=524], D[285+242=527], M[292+241=533], L[290+244=534], D[295+242=537], T[251+329=580],

A[229+366=595], R[411+193=604], D[385+242=627], T[333+329=662], R[500+193=693], C[541+160=701]



Final:



5p 4. The heuristic path algorithm is a best-first search in which the evaluation function is f(n) = (2 − w) g(n) + w h(n). What kind of search does this perform for w = 0, w = 1, and w = 2? For what values is it optimal, assuming that h is admissible?

Definition

An admissible heuristic is one that never overestimates the cost to reach the goal.
E.g.: straight-line distance to Bucharest from Exercise 3. Straight-line distance is admissible because the shortest path between any two points is a straight line, so the straight line cannot be an overestimate.

$w = 0 \Rightarrow f(n) = 2g(n)$ – uniform-cost search, multiplying by 2 does not change node ordering.

$w = 1 \Rightarrow A^*$ search.

$w = 2 \Rightarrow f(n) = 2h(n)$ – greedy best-first search.

$f(n) = (2 - w)\, g(n) + w\, h(n) \Rightarrow f(n) = (2-w)[g(n) + \frac{w}{2-w}\, h(n)] \Rightarrow$
A^* search with heuristic $\frac{w}{2-w}\, h(n)$.

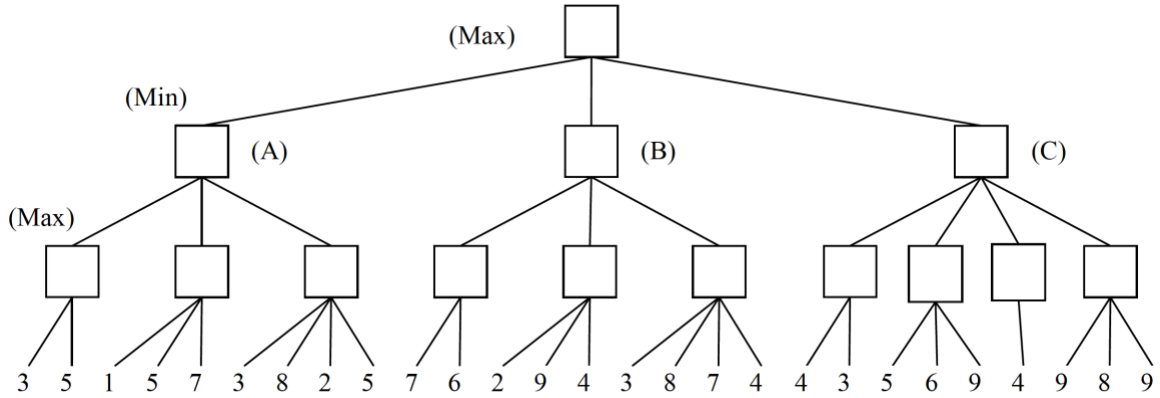
$\forall w \leq 1, \frac{w}{2-w}\, h(n) \leq h(n) \leftarrow$ admissible.
$h(n) =$ admissible (hypothesis).

The algorithm is guaranteed to be optimal for $0 \leq w \leq 1$, since scaling g(n) by a constant does not affect the relative ordering of the chosen paths.

If $w > 1$, it is possible that $wh(n)$ will overestimate the distance to the goal, making the heuristic inadmissible.

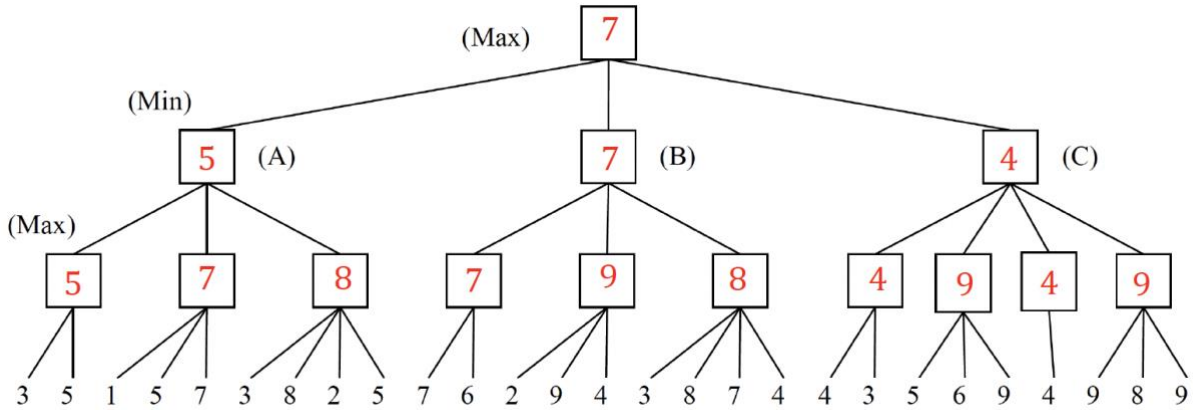
If $w \leq 1$, then the estimate is reduced, but it is still guaranteed to underestimate the distance to the goal state.

5p 5. Given the game tree below, the first player is attempting to maximize their score.



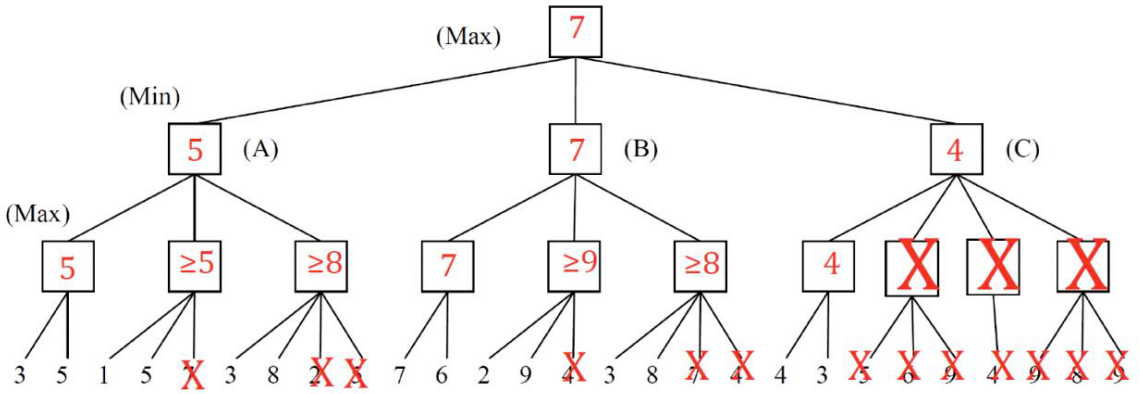
0.5p a. Redraw the figure and fill in each empty box with the value returned by the standard minimax algorithm.
0.5p b. Which is the best starting move for player Max? A, B or C?
2p c. Redraw the figure and fill in each empty box with the value returned by the standard alpha-beta algorithm if the tree were processed in a left-to-right manner. Cross out leaves and (if necessary) nodes that need not be examined.
2p d. Redraw the figure and fill in each empty box with the value returned by the standard alpha-beta algorithm if the tree were processed in a right-to-left manner. Cross out leaves and (if necessary) nodes that need not be examined.

a.

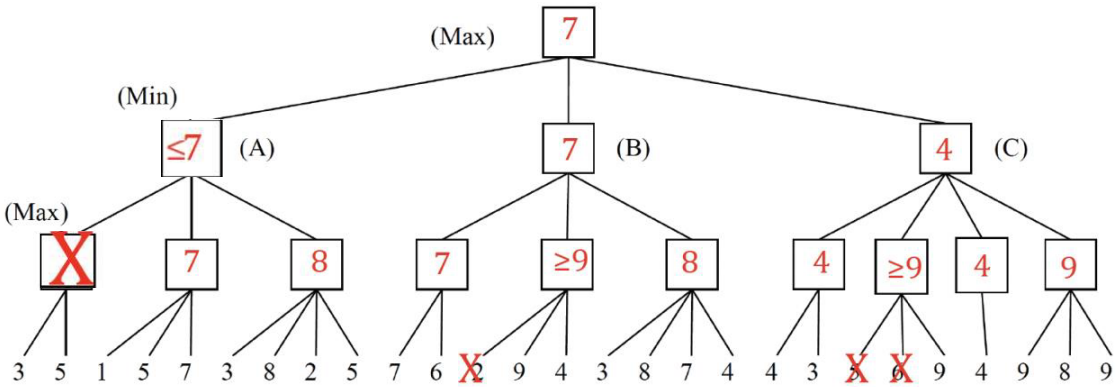


b. B

2p c.


(Max)
7
(Min)
5
(A)
7
(B)
4
(C)
(Max)
5
≥5
≥8
7
≥9
≥8
4
3 5 1 5 7 3 8 2 5 7 6 2 9 4 3 8 7 4 4 3 5 6 9 4 9 8 9

2p d.


(Max)
7
(Min)
≤7
(A)
7
(B)
4
(C)
(Max)
7
8
7
≥9
8
4
≥9
4
9
3 5 1 5 7 3 8 2 5 7 6 2 9 4 3 8 7 4 4 3 5 6 9 4 9 8 9