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 – act humanly**

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 – deterministic**

d. In a zero-sum two player game there is necessarily always a winner. **FALSE - draw**

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. **FALSE – no effect**

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**

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

a. Breadth-first search is a special case of uniform-cost search. **TRUE**

In UCS, node with the lowest path cost is expanded.

In BFS, the shallowest node is expanded.

If all edge costs are equal, UCS becomes same as BFS. Therefore, we can say that BFS is a special case of UCS, where all edge costs are the same.

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

In Best-first search, node with the highest priority (lowest f(n) output) is expanded. F(n) is determined by a heuristic. However, BFS, DFS and UCS do not consider any heuristics. BFS expands the shallowest nodes, while DFS expands the nodes by going as deep as possible. UCS considers only the path cost. So, since they do not use any heuristics, any additional info, they are not considered as types of best-first search.

c. Uniform-cost search is a special case of A∗ search. **TRUE**

A* search uses f(n) = g(n) + h(n), where g(n) = path cost from start to current and h(n) = estimated path cost from current to goal.

UCS uses g(n)

If we say h(n) = 0, f(n) becomes g(n) + 0 = g(n)

Thus, we can say that UCS is a special case of A*, where h(n) = 0.

3. Trace the operation of A∗ search, applied to the problem of reaching Bucharest from Lugoj, using the straight-line distance heuristic.

From	To	g(n)	h(n)	f(n) = g(n) + h(n)
Lugoj	Timisoara	111	329	440
Lugoj	Mehadia	70	241	311
Mehadia	Lugoj	70+70=140	244	384
Mehadia	Drobeta	70+75=145	242	387
Drobeta	Mehadia	145+75=220	241	461
Drobeta	Craiova	145+120=265	160	425
Craiova	Drobeta	265+120=385	242	627
Craiova	Rimnicu	265+146=411	193	604
Craiova	Pitesti	265+138=403	100	503
Timiosara	Arad	111+118=229	366	595
Timisora	Lugoj	111+111=222	244	466
Pitesti	Craiova	403+138=541	160	701
Pitesti	Rimnicu	403+97=500	193	693
Pitesti	**Bucharest**	**403+101=504**	**0**	**504**

Path: **Lugoj → Mehadia → Drobeta → Craiova → Pitesti → Bucharest**

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?

W = 0 → f(n) = 2 g(n)

This is uniform-search algorithm since this algorithm expands the node with the lowest path cost.

W = 1 → f(n) = g(n) + h(n)

This is A* search algorithm.

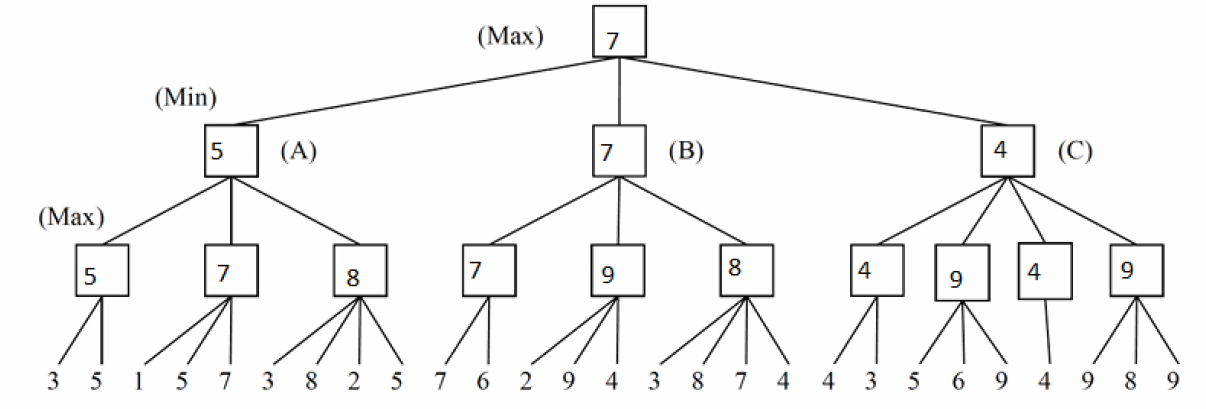
W = 2 → f(n) = 2 h(n)

Since this function only considers heuristics, we can say that this is a type of greedy best-first search algorithm.

The search is optimal when w = 1 if h is admissible. Uniform-cost is not optimal if there are negative cost values, and greedy best-search is not necessarily optimal at all. However, A* is optimal given that h(n) is admissible.

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

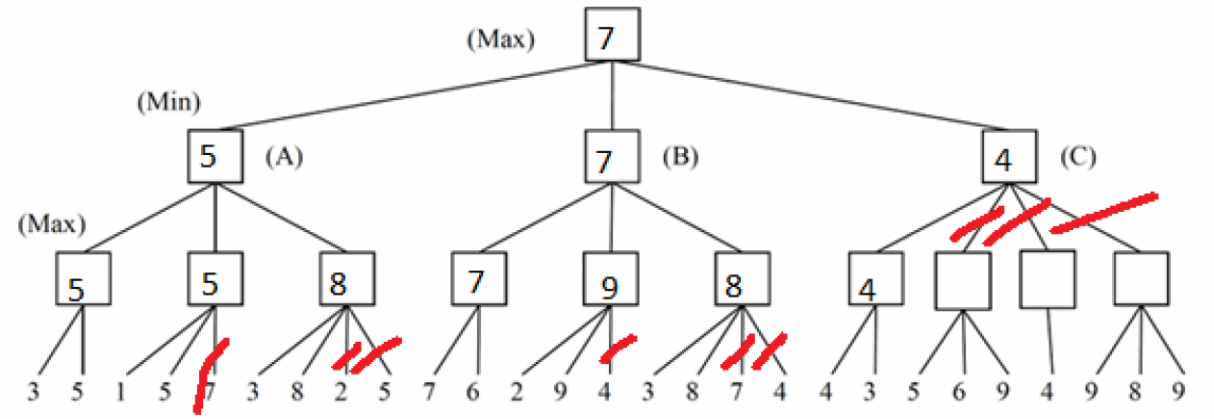
a. Redraw the figure and fill in each empty box with the value returned by the standard minimax algorithm.



b. Which is the best starting move for player Max? A, B or C?

B

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.



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.