



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

DSIC
DEPARTAMENTO DE SISTEMAS
INFORMÁTICOS Y COMPUTACIÓN

Escuela Técnica Superior de Ingeniería Informática



**Departamento de Sistemas Informáticos y Computación
Escuela Técnica Superior de Ingeniería Informática
Universitat Politècnica de València**

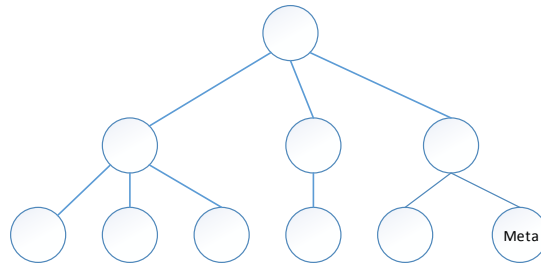
Questions and Problem Statements **INTELLIGENT SYSTEMS**

Block 1: Solving problems by Search

September 2019

MULTIPLE CHOICE QUESTIONS

- 1) Given the space state of the figure, if we apply a DEPTH-FIRST search with *backtracking* (expanding first the leftmost node), which is the maximum number of nodes kept simultaneously in memory (nodes in both OPEN and CLOSED)?



- A. 7
 - B. 5
 - C. 10
 - D. 4
-

- 2) Consider the 8-puzzle problem, where h_1 =number of misplaced tiles (tiles in the wrong place), and h_2 =Manhattan distance. Let $h_3=\min(h_1, h_2)$ and $h_4=\text{abs}(h_1-h_2)$, which of these two heuristic functions, h_3 and h_4 , is admissible?

- A. Only h_3 .
 - B. Only h_4 .
 - C. Both of them.
 - D. None of them.
-

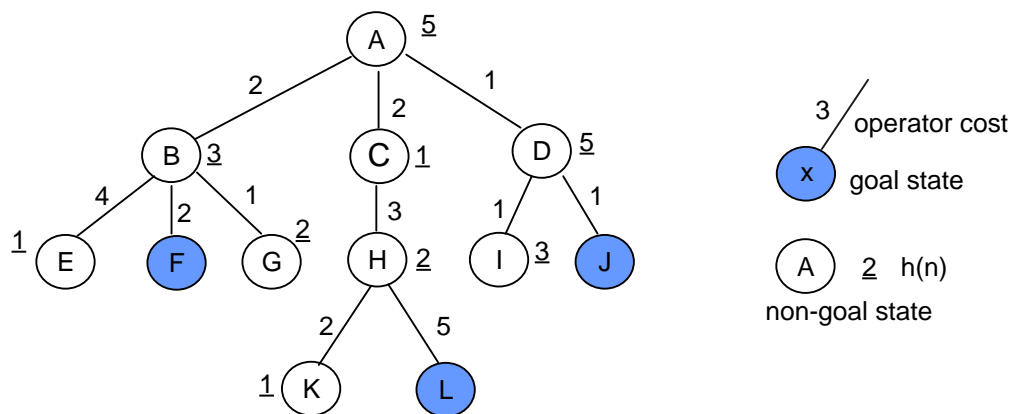
- 3) Consider a search problem with a branching factor $b=10$ and depth of the solution $d=5$. Let's assume we apply a BREADTH-FIRST search (BFS), a DEPTH-FIRST search (DFS) where the maximum depth limit set by the user is 5, and an ITERATIVE DEEPENING algorithm (ID). Regarding the number of generated nodes, which of the following orderings (from the most costly to the least costly) is correct?

- A. $\text{BFS} > \text{ID} > \text{DFS}$
 - B. $\text{BFS} > \text{DFS} > \text{ID}$
 - C. $\text{ID} > \text{BFS} > \text{DFS}$
 - D. $\text{DFS} > \text{BFS} > \text{ID}$
-

- 4) If we use a consistent heuristic function in an A algorithm, which assertion is **FALSE**?

- A. $f(n)$ is a non-decreasing function.
- B. It guarantees the optimal solution in a graph-search version even without re-expanding the nodes.
- C. The search process will never generate a node 'n1' equal to another already generated node 'n2' where $f(n1) < f(n2)$.
- D. The h-value of a child node can be lower than the h-value of a parent node.

- 5) Given the state space of the figure, if we apply a **greedy** search, which of the goals states is first selected as a solution?



- A. J
- B. L
- C. I
- D. F

- 6) Given the state space of figure 5, if we apply an Iterative Deepening (ID) algorithm (expanding first the leftmost node), how many nodes would be generated altogether?

- A. 8
- B. 7
- C. 10
- D. 12

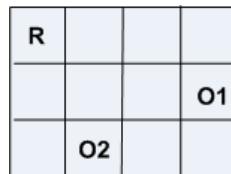
- 7) Given a search problem in which *all the operators have the same cost*, show which of the following assertions is **correct**:

- A. A search algorithm that uses an admissible heuristic will return the shortest solution
- B. A depth-first search (DFS) will always return the lowest cost solution
- C. A breadth-first search (BFS) will return the shortest solution but not the lowest cost solution
- D. A uniform-cost search strategy will return the lowest cost solution but not the shortest solution

8) Given four search algorithms: M1 applies a breadth-first search (BFS); M2 applies a uniform-cost search strategy; M3 applies an A algorithm and uses an admissible heuristics; M4 applies an A algorithm and uses a non-admissible heuristic. Show the assertion that is **incorrect**:

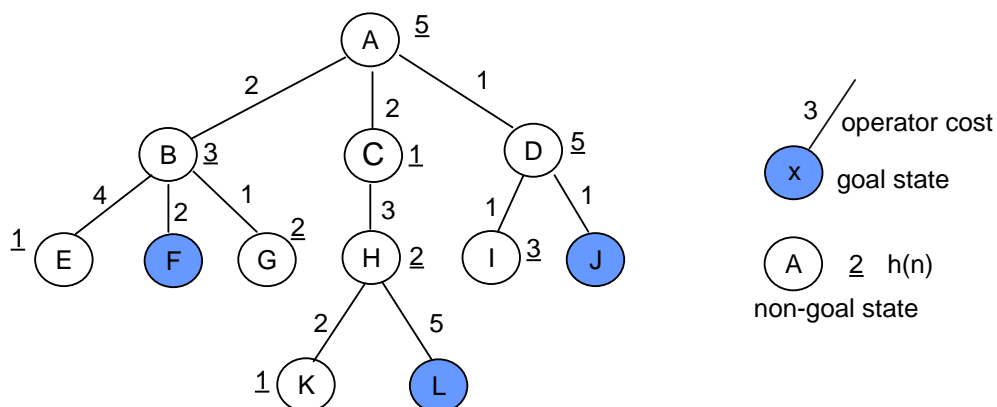
- A. M1, M2 and M3 guarantee the optimal solution regardless the cost of the operators
- B. M2 and M3 guarantee the optimal solution regardless the cost of the operators
- C. M3 will expand fewer nodes than M2
- D. M4 could find the optimal solution

9) The figure below shows a grid where **R** is a robot which wants to reach the position where the object **O1** is located and then reach the position of the object **O2**. The robot only moves horizontally and vertically. **R**, **O1** and **O2** can be placed in any cell of the grid, the figure just shows a particular instance of the problem. Let **n** be a node of a search tree that represents a particular placement of **R**, **O1** and **O2**, and let $\text{manh}(x,y)$ be the Manhattan distance between $x \in y$, where $x, y \in \{R, O1, O2\}$. Which is the **correct** assertion?



- A. $h(n) = \text{manh}(R, O1) + \text{manh}(R, O2)$ is an admissible heuristic for this problem
- B. $h(n) = \text{manh}(R, O1) + \text{manh}(O1, O2)$ is an admissible heuristic for this problem
- C. $h(n) = \text{manh}(R, O1) * 2$ is an admissible heuristic for this problem
- D. It is not possible to define an admissible heuristic for this problem

10) Given the search space of the figure and assuming we apply an algorithm of type A ($f(n) = g(n) + h(n)$), how many nodes need to be generated to find the solution?

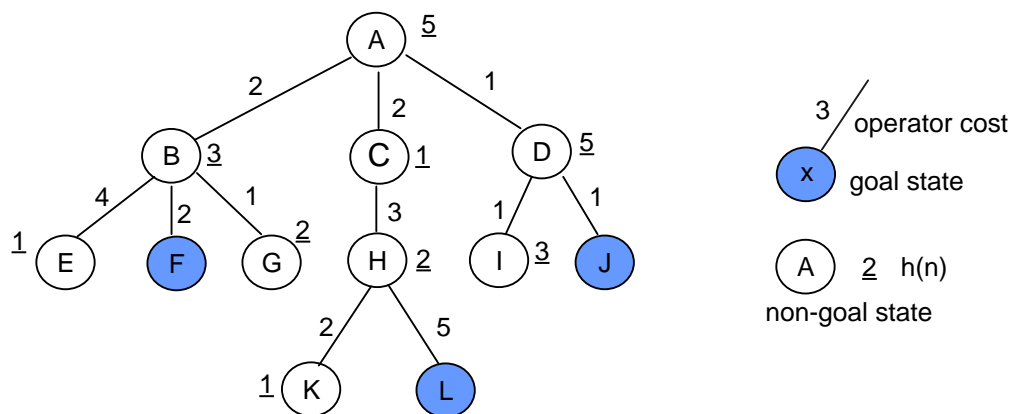


- A. 6
- B. 8
- C. 10
- D. 12

11) Regarding the search space of question 10, mark the **correct** assertion:

- A. The application of the algorithm of type A returns the optimal solution
- B. The function $h(n)$ is consistent (monotone)
- C. A uniform-cost search returns the same solution as the one returned by the algorithm of type A
- D. None of the above.

12) For the search space of the figure and given a search of type A ($f(n)=g(n)+h(n)$), which of the following assertions is **CORRECT**:

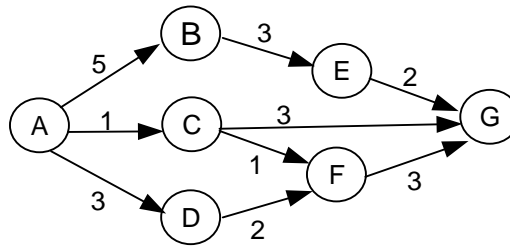


- A. The solution found by the search of type A is node J.
- B. The search of type A generates 10 nodes to find the solution.
- C. The heuristic function $h(n)$ is not admissible.
- D. None of the above.

13) Given four search methods: M1 applies Breadth-first, M2 applies Uniform Cost, M3 applies Depth-first and M4 is an Iterative-Deepening algorithm; assuming all the operators have the same cost, mark the assertion that is **INCORRECT**:

- A. M1, M2, M3 and M4 will find the optimal solution if it exists
- B. M1 and M2 guarantee the optimal solution.
- C. M4 will find the optimal solution.
- D. The memory requirements are bigger for M1 than M4.

14) In the graph below, the numbers on the edges represent the operator cost to go from one node to the other. Mark the assertion that is **CORRECT**.



- A. Breadth-first will find the path A-D-F-G
- B. The cost of the solution found by a Uniform cost algorithm is 5
- C. Breadth-first and Uniform cost will find the same solution
- D. None of the above.

15) If we apply an Iterative Deepening algorithm over the search space of figure 10, how many iterations are necessary to find a solution?

- A. 2
- B. 3
- C. 4
- D. None of the above.

16) Let be a search algorithm of type A ($f(n)=g(n)+h(n)$) where $h(n)$ is admissible and consistent. The algorithm returns a solution path from the initial state A to the goal state G through a node $n1$. Which of the following assertions is **INCORRECT**?:

- A. $f(A) \leq f(n1) \leq f(G)$
- B. $f(G)=h^*(A)$
- C. $h^*(A) < h(n1)$
- D. $f(G)=g(G)$.

17) Let $f1(n)=g(n)+h1(n)$ and $f2(n)=g(n)+h2(n)$ be two evaluation functions for a problem such that $\forall n \ h1(n) \leq h2(n) \leq h^*(n)$. Given a search algorithm of type A that utilizes these functions, mark the assertion that is **TRUE**:

- A. Only one of the two evaluation functions will find the optimal solution
- B. The algorithm that uses $f1(n)$ will expand fewer nodes than the algorithm with $f2(n)$
- C. The algorithm that uses $f1(n)$ will expand more nodes than the algorithm with $f2(n)$
- D. None of the two algorithms will develop a complete search

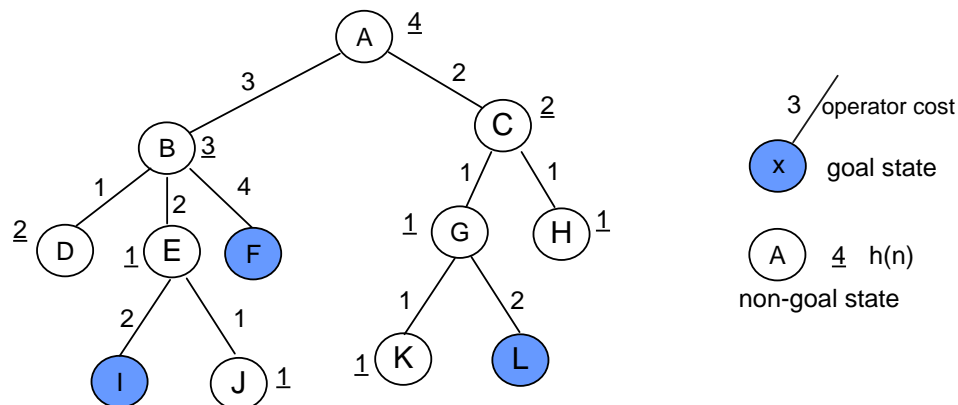
18) Given three search methods: M1 applies a uniform-cost search, M2 is an A* algorithm and M3 applies a greedy search, which assertion is **INCORRECT**?

- A. M1 and M2 will find the optimal-cost solution
- B. There is guarantee that M3 will find the solution more rapidly than M1 and M2
- C. There is no guarantee that M3 will find the optimal-cost solution
- D. M1 will expand more nodes than M2

19) Given the evaluation functions $f_1(n)=g(n)+h_1(n)$ and $f_2(n)=g(n)+h_2(n)$, such that $h_1(n)$ is admissible and $h_2(n)$ is not, mark the CORRECT answer:

- A. Both functions guarantee they will find the optimal-cost solution
- B. There is guarantee that $f_2(n)$ will generate fewer nodes than $f_1(n)$
- C. Only in the case that $h_1(n)$ is a consistent heuristic function, $f_1(n)$ will generate fewer nodes than $f_2(n)$
- D. There exists some node n for which it holds $h_2(n) > h^*(n)$

20) Given the state space of the figure and assuming a breadth-first search (expanding the leftmost node first), which assertion is TRUE?



- A. The algorithm returns node I
- B. The algorithm generates 8 nodes
- C. The algorithm expands 4 nodes
- D. None of the above

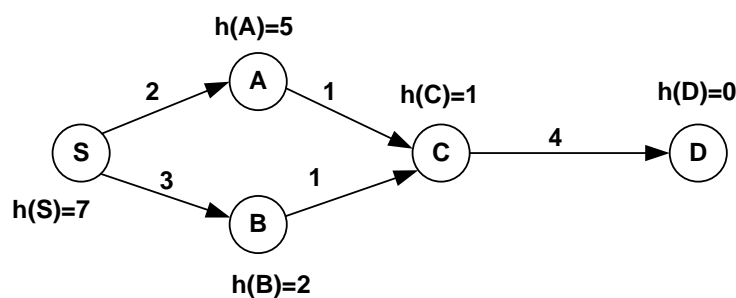
21) Given again the state space of question 20 and assuming now a search algorithm of type A ($f(n)=g(n)+h(n)$), which assertion is FALSE?

- A. $h(n)$ is admissible
- B. The algorithm returns the node L
- C. The algorithm expands 3 nodes
- D. The algorithm generates 7 nodes

22) Assuming that all of the nodes in a search space have more than one child, in which of the following search strategies the order of generation of the nodes is **never** the same as the order of expansion of the nodes?

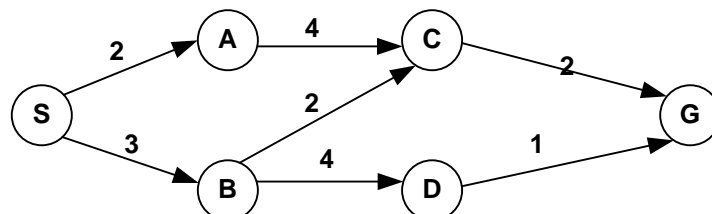
- A. Breadth-first search
- B. Uniform cost search
- C. Depth-first search
- D. Greedy search

23) In the state space of the figure, S is the initial state, D is the goal state, $h(n)$ is the heuristic estimation for each node and the numeric values on the edges are the costs of the arcs. Indicate the **CORRECT** statement.



- A. An A algorithm (TREE SEARCH with control of repeated states in the OPEN list) does not return the optimal solution.
- B. An A algorithm (GRAPH SEARCH with control of repeated states in the CLOSED list such that a newly generated node is discarded if it exists in CLOSED) returns the optimal solution
- C. The answer A is not TRUE since $h(n)$ is not admissible
- D. The answer B is not TRUE since $h(n)$ is not consistent

24) Given the state space of the figure, the number of nodes generated by a Uniform Cost search (TREE SEARCH) is (in case of two nodes with the same f-value, expand first the node that comes alphabetically before):



- A. Higher than the number of nodes of a BFS
- B. Lower than the number of nodes of a BFS
- C. Lower than the number of nodes of a DFS
- D. None of the above answers is TRUE

25) Given a consistent heuristic function $h(n)$ in a search of type $f(n)=g(n)+h(n)$, which of the following assertions is CORRECT?

- A. It does not return the optimal solution
- B. The heuristic value of the parent can be the same as the heuristic value of the child
- C. It never generates a new node $n1$, equal to another already generated node $n2$, such that $f(n1) < f(n2)$.
- D. It never generates a new node which is the same as one already in the CLOSED list

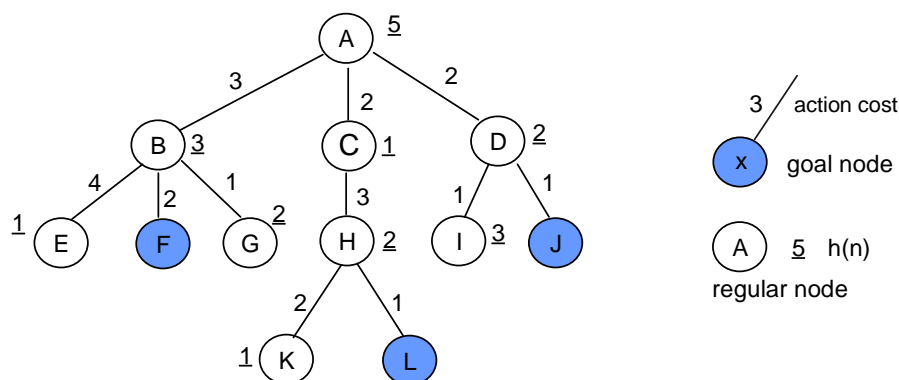
26) Let be a search $f(n)=g(n)+h(n)$, where $h(n)$ is admissible, and two solution nodes $G1$ and $G2$. $G1$ is an optimal solution and $G2$ is not. Let $n1$ be a node that belongs to the solution path to $G1$. Mark the **INCORRECT** statement:

- A. $g(G1) \leq f(G2)$
- B. $f(n1) \leq g(G2)$
- C. $h^*(n1)+g(n1)=f(G1)$
- D. None of the above

27) Regarding the number of nodes in the worst-case scenario for an Iterative Deepening search that finds the solution at level d and a limited Depth-first search with $m=d$ for one same problem, which of the following assertions is CORRECT?

- A. DFS generates more nodes than ID
- B. ID generates more nodes than DFS
- C. DFS and ID generate the same number of nodes
- D. None of the above

28) Given the search space of the figure, if we apply a search of type A ($f(n)=g(n)+h(n)$) how many nodes are generated to find the solution?



- A. 7
- B. 8
- C. 10
- D. 12

29) If we apply a A^* search in CLIPS, rules must not have a `retract` command in the RHS because:

- A. Facts are retracted and so we would not be able to compute the value of $g(n)$
 - B. We would not be able to explore paths other than the path selected at first place
 - C. We would not be able to compute the optimal solution
 - D. None of the above
-

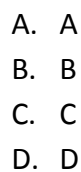
30) Given a search algorithm of type A, $f(n)=g(n)+h(n)$, indicate the **CORRECT** statement:

- A. If $h(n)$ is consistent (and admissible), the algorithm will always expand fewer nodes than an uninformed search
 - B. If $h(n)$ is consistent (and admissible), the algorithm will always expand fewer nodes than when $h(n)$ is not consistent
 - C. Whether or not $h(n)$ is admissible, the same solution will always be found
 - D. None of the above
-

31) We have a search problem where operators have different cost. The application of the GRAPH-SEARCH version of a Uniform Cost algorithm with control of repeated states returns a solution of cost ' c ' at the depth level ' d '. Mark the **CORRECT** answer.

- A. The user does not need to set a maximum depth limit in order to prevent the algorithm from getting stuck in an endless loop
 - B. A solution of cost c' such that $c' > c$ can only be found at a level d' such that $d' > d$.
 - C. The application of an Iterative Deepening algorithm on the same problem will always find the optimal solution.
 - D. The application of a Breadth search on the same problem will always find the optimal solution.
-

32) If we apply an A algorithm over the state space of the figure below, which goal node (shadowy node) will be found?

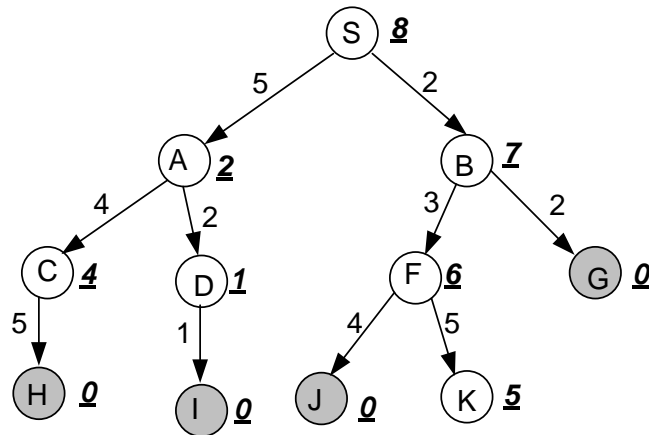


A. Admissible and consistent
B. Admissible and not consistent
C. Non-admissible but consistent
D. Non-admissible and not consistent

- A. It will depend on the effective branching factor of the heuristic function $h(n)$ and the depth of the optimal solution.
- B. It will depend on the cost of the operators.
- C. If $h(n)$ is admissible, the number of nodes will never be higher than the number of nodes generated with a heuristic $h'(n)$ such that $h'(n) = h^*(n)$
- D. None of the above.

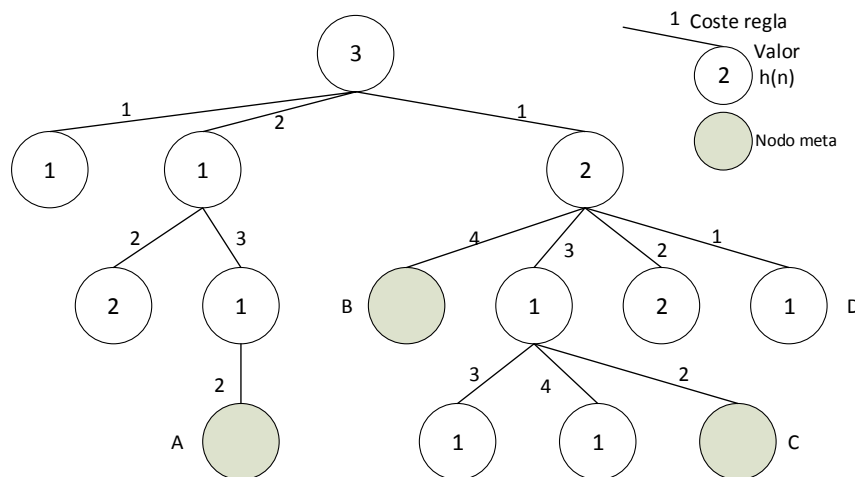
- A. The heuristic function is admissible.
- B. We apply a control of repeated states in the OPEN list.
- C. The heuristic function is consistent.
- D. None of the above.

36) Given the search space of the figure where the value of the edges are the cost of the operators and the number on the right of each node is its heuristic value, mark the **CORRECT** answer:



- A. An algorithm of type A will generate fewer nodes than Uniform Cost
- B. An algorithm of type A will find the optimal solution.
- C. A Breadth search, expanding first the leftmost nodes, will generate the same or a lower number of nodes than Uniform Cost.
- D. None of the above is true.

37) If we apply a *Greedy* search on the search space of the figure, which goal node (shadowy node) will be selected first as a solution and how many nodes are generated to find such a solution?



- A. Goal node A generating 7 nodes
- B. Goal node B generating 8 nodes
- C. Goal node B generating 11 nodes
- D. Goal node C generating 14 nodes

38) Given the search space of the above figure, mark the **INCORRECT** statement:

- A. The heuristic function $h(n)$ is admissible
 - B. The heuristic function $h(n)$ is consistent
 - C. A Breadth-first algorithm will find the same solution as an algorithm of type A
 - D. A Depth-first algorithm will find the same solution as a greedy search algorithm
-

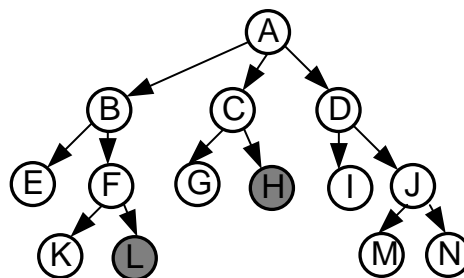
39) Let d_1 , d_2 and d_3 be three depth levels of a search tree where $d_1 < d_2 < d_3$, such that there exists one solution at level d_1 , another solution at level d_2 and another solution at level d_3 (there is only one solution at each level). Mark the **CORRECT** statement:

- A. The time complexity of a Breadth-first algorithm is $O(b^{d_2})$ and the time complexity of an Iterative Deepening algorithm is $O(b^{d_1})$.
 - B. The time complexity of a Depth-first tree search with maximum depth limit $m=d_1$, is $O(b^{d_1+1})$.
 - C. Assuming the user selects $m=d_3$ as maximum depth limit, a Depth-first tree search will always find first the solution at level d_1 or at level d_2 .
 - D. Assuming the user selects $m=d_1$ as maximum depth limit, the time complexity of a Depth-first tree search and an Iterative Deepening algorithm is $O(b^{d_1})$ for both.
-

40) Let's assume an algorithm A^* is applied to solve a problem and that G is the solution node found. Show the **FALSE** statement:

- A. If $h(n)$ is consistent then $\forall n_1, n_2$ such that n_2 is a child node of n_1 , it always holds $h(n_2) \geq h(n_1)$
 - B. $\forall n_1, n_2$, such that n_1 and n_2 are two nodes on the optimal path to G , it always holds $g(n_1) + h^*(n_1) = g(n_2) + h^*(n_2)$
 - C. $\forall n$, such that n is a node on the optimal path to G , it always holds $f(n) \leq g(G)$
 - D. It always holds $f(G) = g(G)$.
-

41) Let be the search tree of the figure, mark the answer that shows the correct order in which nodes would be generated when applying Iterative Deepening and the reached goal node.

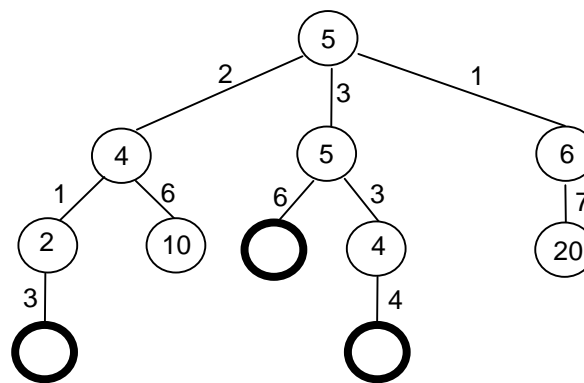


- A. ABCDEFKL and finds goal state L
- B. ABCDEFGHIJKL and finds goal state H
- C. AABCDABCDEFHG and finds goal state H
- D. ABCDEFGH and finds goal state H

42) Given two A* algorithms for one same problem, where algorithm A1 uses heuristic $h_1(n)$ and algorithm A2 uses $h_2(n)$, such that $\forall n, h^*(n) \geq h_2(n) > h_1(n)$, show the **CORRECT** answer:

- A. It is guaranteed that A1 will take less time than A2
- B. It is guaranteed that A1 will expand fewer nodes than A2
- C. The solution found by A2 will be better than the one found by A1
- D. None of the above answers is correct

43) Let be the search tree of the figure, where bold-circled nodes are goal states, the value inside a node is the heuristic value of the node and the numeric value on the arcs is the operator cost. Show the **CORRECT** answer:



- A. The heuristic is admissible and consistent
- B. The heuristic is not admissible nor consistent
- C. The optimal solution is found when an algorithm of type A is applied to the tree
- D. None of the above answers is correct

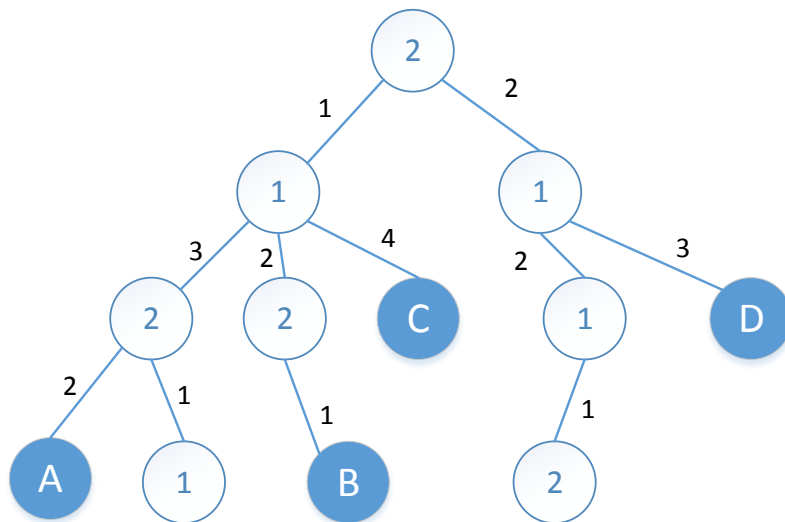
44) Let be a search problem where operators have different costs. We know the search tree contains a solution node G_1 at level d_1 , and a solution node G_2 , which is an optimal solution, at level d_2 . We also know that $d_2 > d_1$. Show the **CORRECT** answer:

- A. The time complexity of a breadth-first strategy with respect to the number of generated nodes is $O(b^{d_1})$
- B. A depth-first strategy will never return the solution G_1
- C. An iterative deepening strategy will never find the solution G_1
- D. A uniform-cost strategy will always find the solution G_2

45) Let be the search tree generated with an A* algorithm. The tree contains two nodes, n_1 and n_2 , which correspond to two repeated states. We also know that n_1 is a node on the optimal path to a solution node G , whereas n_2 is not on the optimal path to G . Show the **INCORRECT** answer:

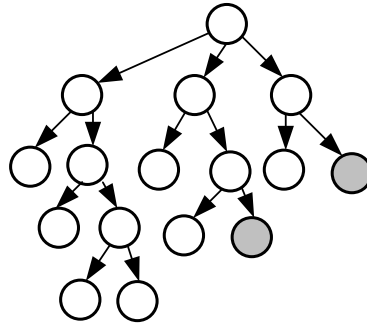
- A. Se cumple siempre $f(n_1) \leq f(G)$
- B. Se cumple siempre $g(n_1) < g(n_2)$
- C. Se cumple siempre $h(n_1) < h(n_2)$
- D. Se cumple siempre $h(n_2) \leq h^*(n_2)$

46) Let be the search tree of the figure, where the value inside the node is the heuristic value of the node and the numeric value on the arcs is the operator cost. The nodes which are labeled as A, B, C and D denote goal nodes. If we apply an A-type search strategy, which goal node is found at first place?



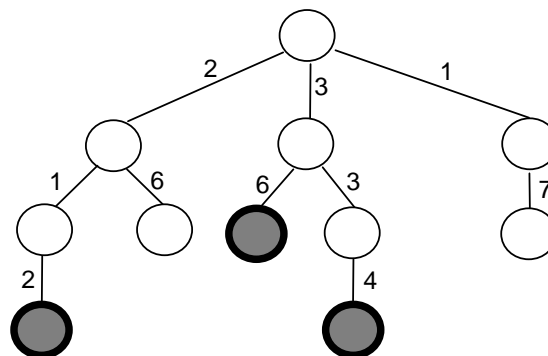
- A. A
- B. B
- C. C
- D. D

47) If we apply an Iterative Deepening (ID) search strategy to the tree shown below, which is the maximum number of nodes kept in memory? (Assume we expand first the leftmost node among the nodes that are at the same depth level)



- A. 6
- B. 3
- C. 4
- D. 5

48) The figure shows a search tree where the shadowy nodes are goal nodes. Show the **CORRECT** answer:

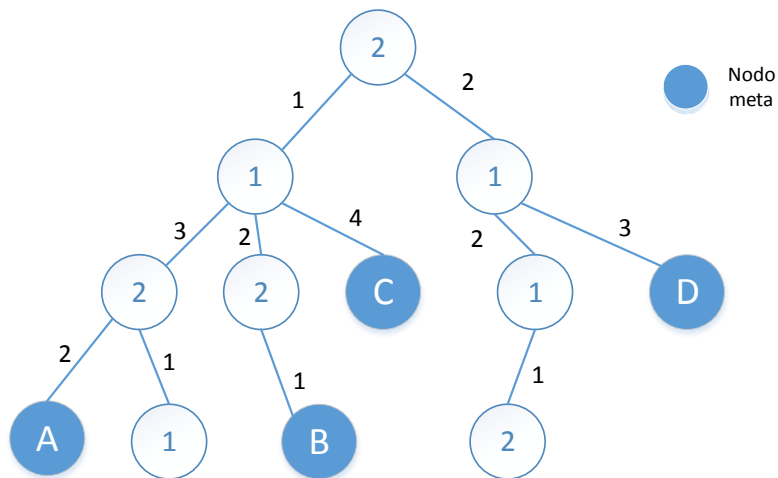


- A. The application of a Breadth-first search returns the same solution node as the application of Uniform-cost search.
- B. The application of a Breadth-first search returns the same solution node as the application of Depth-first search with depth limit $m=2$.
- C. The application of a Breadth-first search returns the same solution node as the application of Depth-first search with depth limit $m=3$.
- D. The application of a Uniform-cost search returns the same solution node as the application of Iterative Deepening search.

49) The application of an admissible heuristic, h_1 , to a problem returns a solution node G_1 and the number of expanded nodes is n_1 . The application of an admissible heuristic, h_2 , to the same problem, where h_2 dominates h_1 , returns a solution node G_2 and expands n_2 nodes. Mark the **CORRECT** answer:

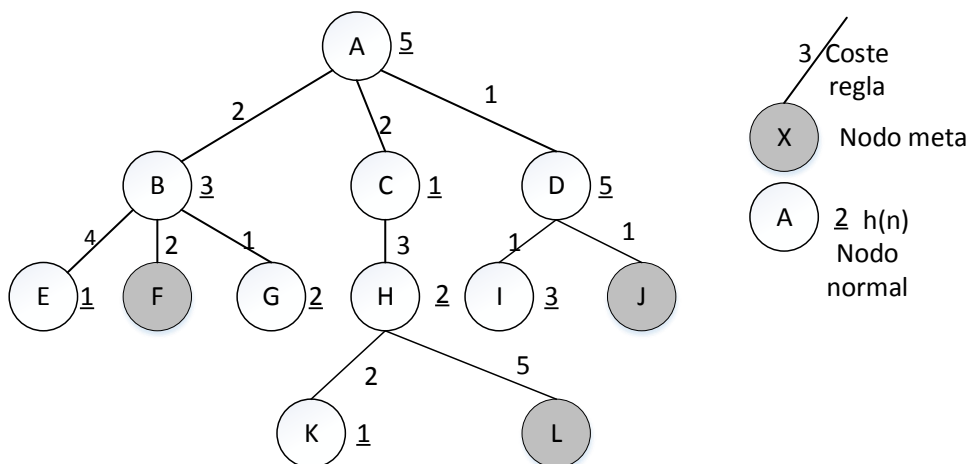
- A. It holds that $g(G_1) < g(G_2)$
- B. It holds that $h_1(G_1) < h_2(G_2)$
- C. It holds that $n_1 < n_2$
- D. None of the above answers is correct.

50) Given the below tree, how many nodes would be generated (including the root node) if we apply an A algorithm? (in case of nodes with the same value of $f(n)$, expand the leftmost node).



- A. 6
- B. 8
- C. 9
- D. 10

51) Given the state space of the figure, if we apply a greedy search, which is the goal node (shadowy node) that will be returned as first solution? (in case of nodes with the same value of $f(n)$, expand the leftmost nodes)



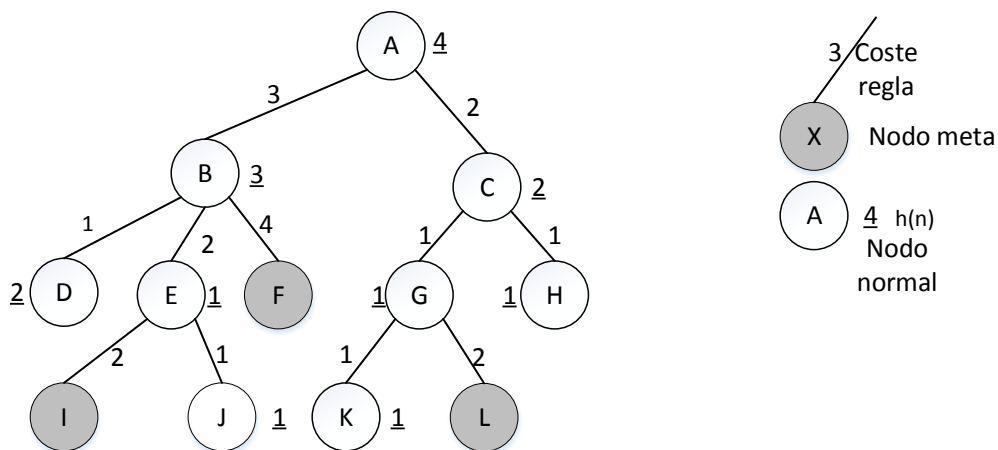
- A. L
- B. J
- C. I
- D. F

52) Given two evaluation functions $f1(n)=g(n)+h1(n)$ and $f2(n)=g(n)+h2(n)$, such that $h1(n)$ is admissible and $h2(n)$ is not admissible. Show the **CORRECT** answer:

Exercises Block 1- Search

- A. If both functions are used in an algorithm of type A, they guarantee the optimal solution
- B. Only in the case that $h_1(n)$ is a consistent heuristic, $f_1(n)$ will generate a smaller search space than $f_2(n)$
- C. It exists a node n for which it holds $h_2(n) > h^*(n)$
- D. We can affirm that $f_2(n)$ guarantees to find a smaller search space than $f_1(n)$

53) Assuming we apply a search of type A ($f(n)=g(n)+h(n)$), in the state space of the figure, which of the following assertions is **FALSE?**:



- A. $h(n)$ is admissible
- B. The search process expands 3 nodes
- C. The search process generates 7 nodes
- D. The solution returned is node L

54) Given a search problem where all operators have the same cost, mark the **CORRECT** statement:

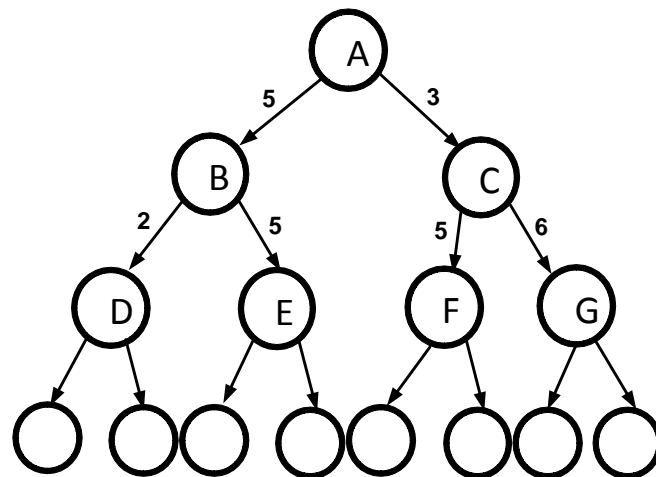
- A. An algorithm of type A that uses an admissible heuristic will return the shortest solution
- B. A breadth-first strategy will return the shortest solution but not the least cost solution
- C. A uniform cost strategy will return the least cost solution but not the shortest solution
- D. A depth-first search strategy will always return the least cost solution

EXERCISES (open answer questions)

Exercise 1

Given the search space in the figure, say the order in which nodes will be expanded according to the following search strategies. Leaf nodes are not expanded. Values in the branches represent the action cost. The heuristic values of the nodes are as follows: $h(B)=17$, $h(C)=14$, $h(D)=13$, $h(E)=11$, $h(F)=10$, $h(G)=7$. Show the OPEN and CLOSED lists.

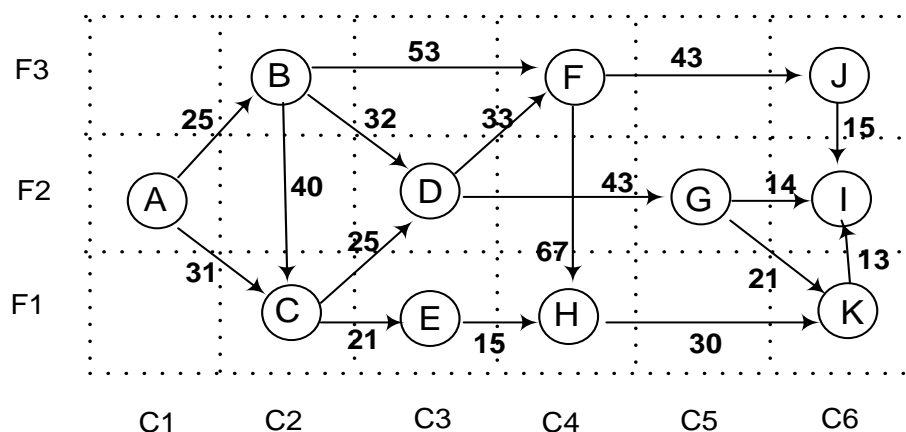
- Breadth-first
- Uniform cost
- Depth-first (expands the leftmost node first)
- Iterative deepening (one depth-level deeper at each iteration)
- Best-first (greedy)
- A algorithm



Exercise 2

In the graph below, nodes represent cities and the edge labels are the real distance (in km) between pairs of cities. The graph is displayed in a map where each city is placed in a quadrant map. The initial state is A and the goal state is I.

Let's suppose we use the heuristic Manhattan-distance (same application as in the 8-puzzle problem from a starting node to a final node through the quadrants in the map; for example, the Manhattan-distance between city A and city E is 3; the Manhattan-distance between city E and I is 4). Let $h(n) = \text{Distancias_Manhattan}(n) * 10$; the operators cost is the real distance in km. between pair of cities (edges labels). Answer the following questions by providing **APPROPRIATE JUSTIFICATIONS** to all your answers. Show clearly the creation of the OPEN list in each question.



- 1) We want to find a solution to go from city A to city I that traverses the minimum possible number of cities. Pick the appropriate strategy and draw the tree to find the solution path. In case of two nodes with the same value of the evaluation function, expand first the node that comes alphabetically before. Avoid repeated states.
- 2) We want to find a solution to go from city A to city I. Show the search tree that results from applying an algorithm type A ($f(n)=g(n)+h(n)$). Which solution will this algorithm find? Is it an A* search? Why? Avoid repeated states.
- 3) Consider again the same problem of going from city A to city I. Show the search tree which results from applying a depth-first expansion up to maximum depth $m=4$. Assume that nodes expanded first are those that come alphabetically before. Which solution will this search find? How many nodes are generated and how many expanded? Avoid repeated states (discard deeper nodes).
- 4) Consider now the problem of going from city A to any city in column C6. Show the search tree which results from applying a uniform cost search. Which city in C6 is reached first? What is the solution cost? How many nodes are generated and how many nodes are expanded? Avoid repeated states.
- 5) Consider now the problem of going from city A to any city in column C6. Show the search tree which results from applying a greedy best-first search. Which city in C6 is reached first? What is the solution cost? Avoid repeated states.

Exercise 3

We wish to find the route followed by a robot (R) to pick up two objects located in a room (O1 and O2) as indicated in the figure. The robot can only move vertically and horizontally. In each movement it only moves one square. Whenever the robot arrives to a square that contains an object, he takes it. The cost of each movement is 1.

R			
			O1
	O2		

Let $h(n)=\sum d_j$ be a heuristic function where d_j is the distance in movements of the robot to the object 'j'. That is, the heuristic function is the sum of the Manhattan distance of the robot 'R' to each remaining object in the room. Consider that, when the robot takes an object, the distance to that object becomes 0 and that the object moves with the robot once collected. The cost of each movement is 1.

- 1) Draw the search tree resulting of applying a breadth-first algorithm. Depict only the first three levels. Avoid repeated states.
- 2) Draw the search tree resulting of applying an A-algorithm ($f(n)=g(n)+h(n)$) to find the route the robot must follow to gather the two objects. Show clearly the value $f(n)$ of each node, the order of the node expansion, and the OPEN list at each iteration. Which solution, and with which cost, is found with this algorithm? The heuristic function 'h', is it an A* heuristic? Why? Justify your answers.
- 3) Following the tree of section 2, the application of a greedy search, which solution will it find? Is it an optimal solution? Why? Justify your answers.

NOTES for the node expansion:

- Note 1: Apply movements in this order to create the search trees: first, move UP; second, move DOWN; third, move RIGHT; finally, move LEFT.
- Note 2: If two nodes have the same f -value, expand first the **deepest** node. If both are at the same level, expand first the **oldest** node.
- Note 3: Avoid repeated nodes.

Exercise 4

Given two admissible heuristic functions (h_1, h_2), compare the following compound functions and respond which of them will generate the lowest number of nodes: (i) $\max(h_1, h_2)$, (ii) $\min(h_1, h_2)$, (iii) $h_1 + h_2$.

Exercise 5

Consider an admissible A* algorithm with $f(n) = g(n) + h(n)$ where the cost of each operator is 1. If $g(n)$ changes and operators have now different cost (non-uniform cost), is the algorithm still A*?

Exercise 6

In order to solve a problem as a search process we have used three different operators O1, O2 and O3 whose costs are 10, 20 and 30 respectively. The search program asks the user for the maximum depth level of the tree expansion and the program returns the level where the solution has been found, the number of generated nodes and the solution cost. We know that for a maximum depth level equal to five ($D=5$), breadth-first strategy has found a solution at depth level 2 and the cost of the solution is 50 (there only exists a solution at level 2). According to these data:

- Is the found solution the optimal one?. Otherwise, which strategy would you use to find the optimal solution?
- If the user selects $D=8$ and breadth-first strategy, what can you say about the solution returned by the program? Compare this new solution with the one presented above in the statement.
- If the user selects $D=8$ and iterative-deepening strategy, what can you say about the solution returned by the program? Compare this new solution with the one presented above in the statement.

Exercise 7

A particular search problem is solved with the following search strategies: breadth-first, depth-first, an A* algorithm with $h_1(n)$, another A* algorithm with $h_2(n) > h_1(n)$ and a non-A* algorithm. The search program asks the user to select the depth level for the search tree and the obtained results are those shown in the table. Explain and reason which search strategy has been executed for the results obtained at each row in the table.

User depth level	Sol. depth level	Number of nodes
15	13	264
18	13	860
18	18	565
20	13	325
25	20	205

Exercise 8

We have a search problem whose operators have different cost (non-uniform cost). If we apply an Iterative Deepening (ID) search strategy to solve the problem, what can you say about the solution found?. Explain if ID will find a solution and if so define the solution found in terms of optimality, time complexity, space complexity, etc.

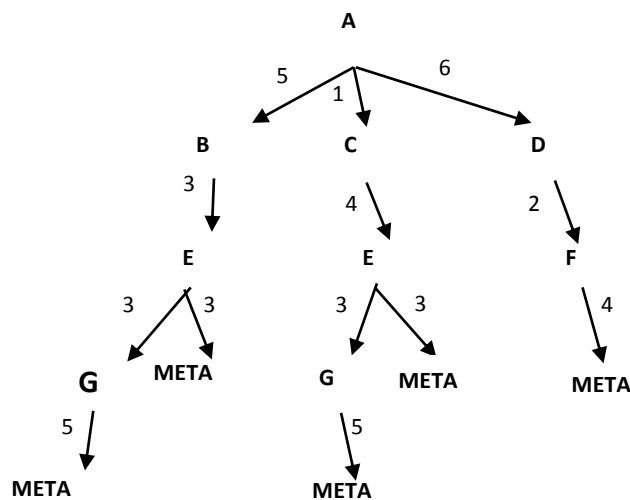
Exercise 9

Given the following search tree where values in the edges represent the cost of applying operators and the values of $h(n)$ for each node are:

A	B	C	D	E	F	G
5	4	3	5	2	1	3

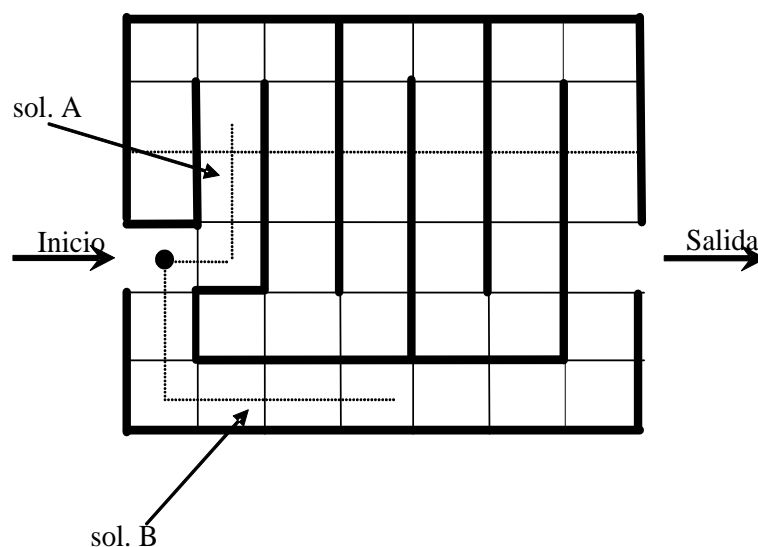
Show the search process if we apply:

- A greedy best-first method
- An algorithm A
- According to the given values, is it an A* search?



Exercise 10

Consider the maze in the figure where the goal is to find the finish location (Salida) starting from initial location (Inicio). The possible operators are UP, DOWN, LEFT and RIGHT which are only valid if there is not a thick line (bold line = obstacle) between the grid boxed in the map maze. As you can see there are two solutions for this problem (Sol. A and Sol. B).



Consider that the heuristic used is Manhattan-distance (same application as in the 8-puzzle problem from the Inicio to the Salida). We assume a unit cost for each move. We assume the search process avoids direct loops. Answer the following questions by providing **APPROPRIATE JUSTIFICATIONS** to all your answers.

- 1) Draw the search tree that results from applying an algorithm type A ($f(n)=g(n)+h(n)$) and number the 7 first expanded nodes (squares) in this maze. Which solution will this algorithm find? Is it an A* search? Show your reasoning.
- 2) If operator RIGHT costs 2 units instead of one unit, the solution found by the algorithm A in section 1 would be the same? same solution quality and same search cost? Why?
- 3) If we apply a greedy search, which solution is found? Justify your answer.
- 4) If we apply a breadth-first search, which solution is found? Justify your answer.
- 5) Assume we want to apply a depth-first search. Which parameters would we have to set up to run a depth-first search that guarantees to find a solution? Indicate which solution depth-first would find under those parameters.

Exercise 11

Given the following search methods in graphs:

- M1) A* algorithm, with $f(n)=g(n)+h_1(n)$
- M2) A* algorithm, with $f(n)=g(n)+h_2(n)$ / $h_2(n) > h_1(n)$
- M3) A algorithm, non A*

give an answer to the these questions:

- a) Which method will expand more nodes?
 - M1 or M2?
 - M1 or M3?
- b) Say, if possible, which method will return the best quality solution:
 - M1 or M2?
 - M1 or M3?
- c) About the total search cost, make a comparison between:
 - M1 and M2
 - M1 and M3

Exercise 12

The tree below represents the search space for a particular problem. Values inside the nodes are the result of applying $f(n)=g(n)+h(n)$, $h(n) \leq h^*(n)$. All actions cost are equal (1 unit cost). The two nodes in bold are goal states, and the nodes crossed out mean that no children are generated. Answer the following questions:

- a) Which solution will be found if we apply the 'f' function? Which is the cost of the solution path? How many nodes have been generated to find such a solution path?
- b) According to the f-values of each node, is the 'h' function consistent (A* monotone)? Why? Justify your answer.

Exercise 14

We have two A* algorithms for the same problem with heuristic functions $h_1(n)$ and $h_2(n)$ respectively.

- a) Assuming, $\forall n \ h_1(n) \leq h_2(n)$, which algorithm will find the best quality solution? Which method will generate the lowest number of nodes?
- b) Assuming $h_1(n)$ is A* monotone (consistent) and $h_2(n)$ is not, which method will have the lowest branching factor?

Exercise 15

Answer the following questions:

- a) Consider a state space where the initial state is the number 1 and the successor function (operator) for any state returns two children: $2n$ and $2n+1$.
 1. Draw the state space for states from 1 to 15
 2. Assuming that the goal state is number 11; depict the search space and the order in which nodes are expanded if we apply a breadth-first algorithm and an iterative-deepening algorithm.
- b) Given the evaluation function $f(n) = (2-a).g(n) + h(n)$ such that $h(n) \leq h^*$, which values would variable 'a' take on if we wish to guarantee the optimal solution? Do you think the algorithm will find the optimal solution if $a=2$?

Exercise 16

Given the following A algorithms where $h^*(n)$ represents the cost from node n to the optimal state:

- 1) $f_1(n) = g(n) + h_1(n) \ / \ h_1(n) \leq h^*(n)$
- 2) $f_2(n) = g(n) + h_2(n) \ / \ h_2(n) > h_1(n)$
- 3) $f_3(n) = g(n) + h_3(n) \ / \ h_1(n) \leq (h_3(n) \leq (1+a) h^*(n))$, being a a positive constant.
- 4) $f_4(n) = g(n) + h_4(n) \ / \ h_4(n) \leq h_1(n)$

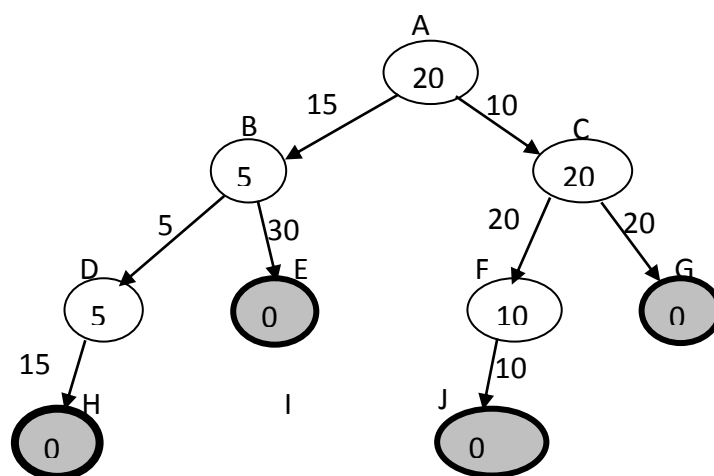
Respond briefly to the following questions:

- a) Which of the above methods is admissible?
- b) Which of the two methods, f_1 or f_4 , will have a larger profoundness?
- c) What can you say about the quality of the solution returned by $f_3(n)$? And about the solution returned by $f_2(n)$?
- d) Given a very complex problem, which method is the most appropriate in each of the following cases:
 - d.1) When we want a response in a VERY SHORT TIME. In this case, importance is given to promptness rather than the solution quality or even the guarantee to eventually find a solution.
 - d.2) When we want a response in a SHORT TIME and with a good quality
 - d.3) When we want the optimal solution
- e) In general, is it possible to determine for a given problem ...
 - e.1) the value of $h^*(n)$?
 - e.2) if an heuristic function satisfies the condition $h(n) \leq h^*(n)$?
 - e.3) if an heuristic function satisfies the monotonicity condition?

Exercise 17

The following tree represents the complete search space of a particular problem. Edges are labelled with the cost of operators ($g(n_i \rightarrow n_j)$). Values inside the nodes are the estimations to a goal state ($h(n)$). Shaded nodes represent goal states. Apply an A-type algorithm from the initial node (A) and indicate clearly the order of expansion of nodes (for instance, step i: node X is expanded and nodes Y and Z are generated). Answer the following questions:

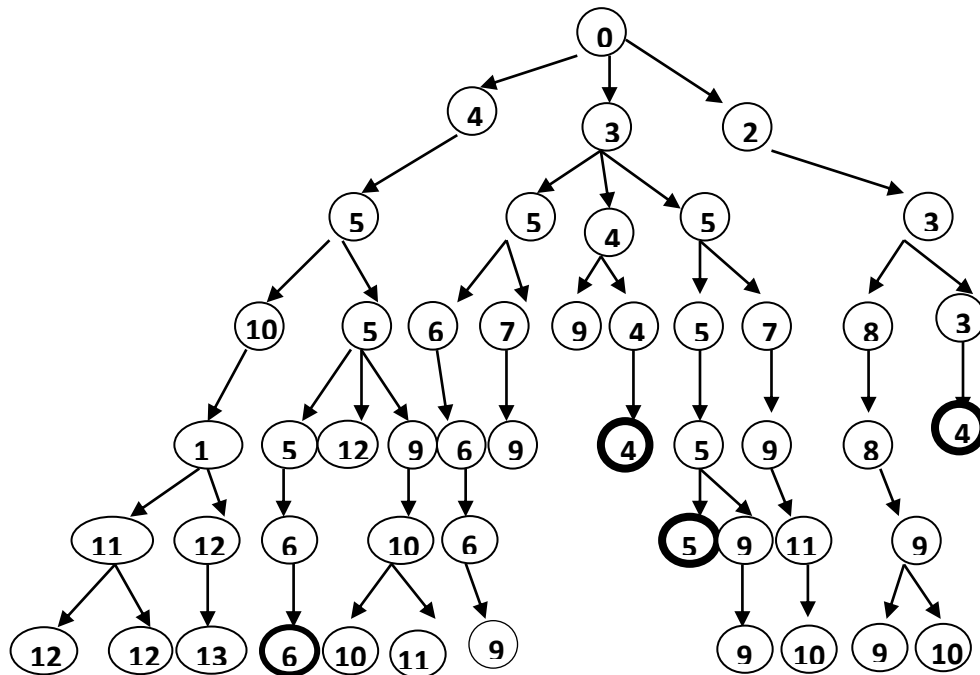
- Is this an A* search? Why?
- Show the goal state returned by the search process
- Does the algorithm return the optimal solution? Why?
- Is the $f(n)=g(n)+h(n)$ a monotone function? Why?
- Apply a greedy best-first search indicating the order of expansion of nodes and the goal state that would be reached.



Exercise 18

The following search tree represents the solution state space of a particular problem. Values in each node are the result of applying a function $f(n)=g(n)+h(n)$, $h(n) \leq h^*(n)$ (admissible heuristic or A*); the cost of all operators is 1. Nodes circled in bold represent solution states for the problem. Answer the following questions:

- If we apply a heuristic search with function $f(n)$, which solution would this method find? how many nodes would be necessary generate to find such a solution?
- If we apply an iterative deepening search strategy, which solution would this method find? Is this the same solution as the one found in section a)? Otherwise, show which solution is better.
- If we apply a depth-first search, which solution would this method find?
- If we apply a heuristic search with function $f_1(n)=g(n)+h_1(n)$, such that $h(n) < h_1(n) \leq h^*(n)$, is the solution path found with $f_1(n)$ better/worse/the same as the one obtained with $f(n)$? Which evaluation function generates the minimum number of nodes?

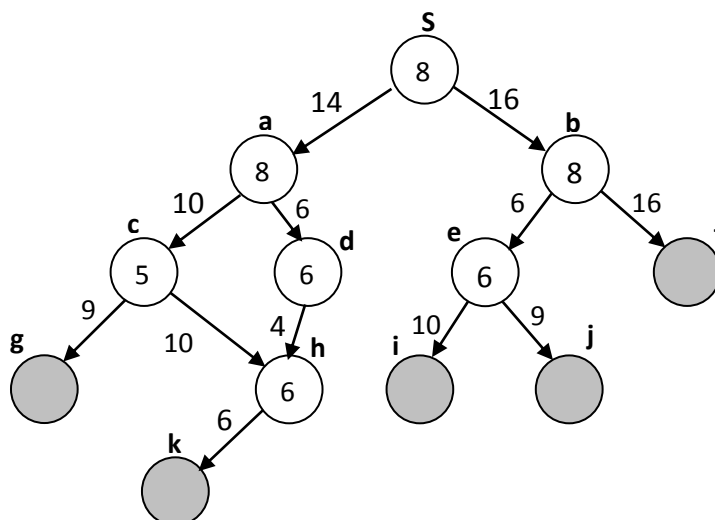


Exercise 19

The figure below shows a search process in a state space where the values labelling the edges are the cost of operators ($g(n)$), the values inside the nodes are the heuristic estimation ($h(n)$), and the shadowed nodes are the goal nodes. Over this search space apply i) **an A algorithm** and ii) a **uniform cost search process** and answer the following questions for each of these two methods:

- The order in which nodes are expanded and which of the goal states is reached
- Does the strategy find the optimal solution? Why?

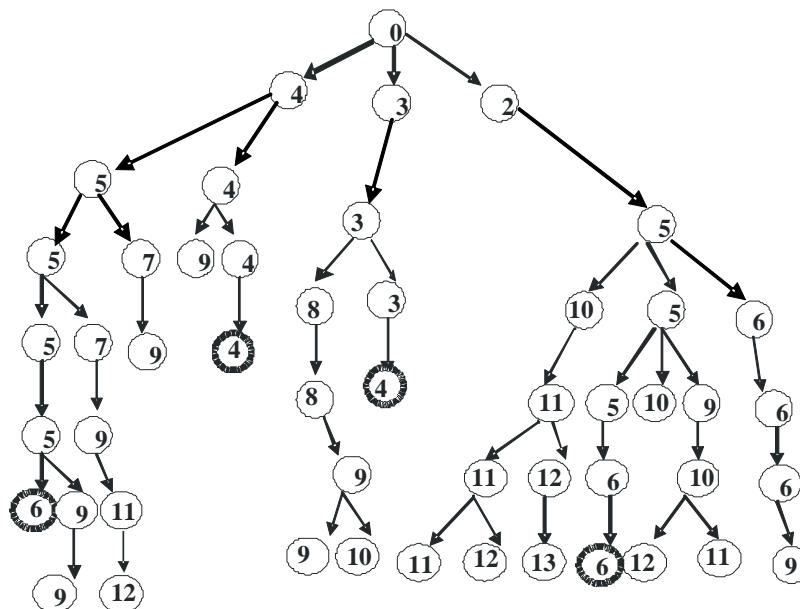
In case of two nodes with the same value of the evaluation function, expand first the node at the shallowest level



Exercise 20

The tree below represents the search space for a particular problem. Values inside the nodes are the result of applying the function $f(n)=g(n)+h(n)$, where $h(n)$ is a heuristic function such that $h(n) \leq h^*(n)$. All actions have the same cost (1). Nodes in bold are goal states. Answer the following questions:

- Which solution will the A* algorithm find by applying the 'f' function? How many nodes are generated to find the solution path?
- If we apply an iterative deepening strategy in the search tree, which solution is found? Is this solution better than the one found by A*? Why?
- If we use an evaluation function $f_1(n)=g(n)+h_1(n)$, such that $h(n) < h_1(n) \leq h^*(n)$, the solution found with 'f1', is it better/worse/equal to the solution found with function 'f'? which strategy will require less search cost?
- If we apply a uniform-cost search, which solution is found? How many nodes have to be generated to find a solution? Is this solution better or worse than the solution found in question a)? Why?



Exercise 21

The figure (a) shows the initial state of a sliding tile puzzle that consists of a row of 5 squares, with 2 white tiles (W) at the left, 2 black tiles (B) at the right and an empty space in the middle. The figure (b) shows the final state we want to reach.

1	2	3	4	5
W	W		B	B

(a) Initial state

1	2	3	4	5
B	B		W	W

(b) Final state

The legal moves or actions, with costs, are as follows:

- If the empty space is next to a tile, the tile may move into the empty space (cost=1)
- A tile may hop over 1 tile (of either colour) into the empty space (cost=1)
- A tile may hop over 2 tiles (of either colour) into the empty space (cost=2)

Additionally, tiles W can only move towards the RIGHT, and tiles B can only move towards the LEFT.

Consider the following heuristic based on the distance of a tile to its “finishing zone”. We define the “finishing zone” of a tile to the two squares into one of which that tile has to move; that is, the finishing zone for each of the W tiles are the squares at positions 4 and 5, and the finishing zone of the B tiles are the squares 1 and 2. The heuristic function ‘h’ applied over a tile that is at one position (1, 2, 3, 4 or 5) returns an estimation of moving that tile to its finishing zone. The h-values for each tile and position are:

POSITIONS	1	2	3	4	5
function $h(W_i)$:	2	1	1	0	0
function $h(B_i)$:	0	0	1	1	2

For example, the estimation of a tile W that is at position 1 is 2 ($h(W_1)=2$), or the h-value of a tile B that is at position 1 is 0 because the tile is already in its finishing zone ($h(B_1)=0$). The application of the h-function over a problem state is the sum of the heuristic values for each of the tiles in that state. For instance, given the state WW_BB, the result of applying the heuristic function is:

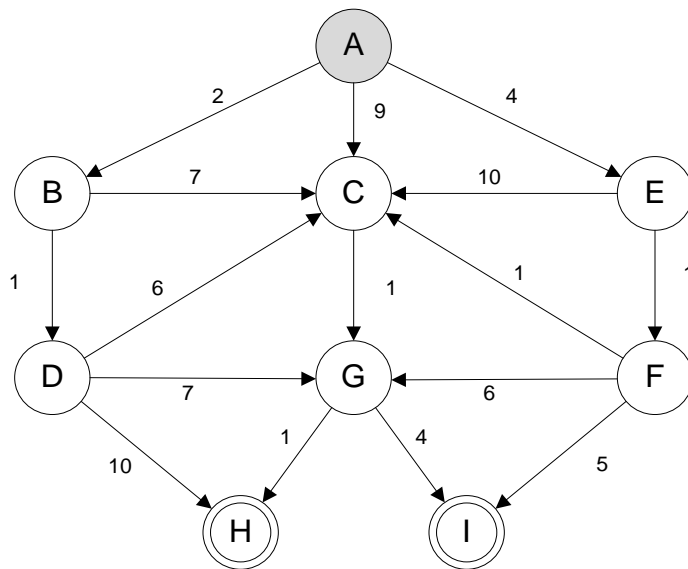
$$h(WW_BB)=h(W_1)+h(W_2)+h(B_4)+h(B_5)=2+1+1+2=6$$

Answer the following questions by providing **APPROPRIATE JUSTIFICATIONS** to all your answers.

- 1) Draw the search tree resulting of applying an A-algorithm ($f(n)=g(n)+h(n)$). Show clearly the value $f(n)$ of each node, the order of nodes expansion and the OPEN list at each iteration. Which solution, and with which cost, will this algorithm find? Is it an A* search? Why?
 - a. *Note 1: Apply actions in this order to create the search tree: first, action A1 over a W tile; then A2 over a W tile; then A3 over a W tile; then A1 over a B tile; then A2 over a B tile and, finally, A3 over a B tile.*
 - b. *Note 2: If two nodes have the same f-value, expand first the **deepest** node. If both are at the same level, expand first the **oldest** node.*
 - c. *Note 3: Avoid repeated nodes.*
- 2) Consider the application of a greedy best-first search. Which solution will it find? Is it an optimal solution? Why?
- 3) In case of a depth-first expansion, is it necessary to set a maximum depth level of the tree to guarantee a solution is found? Why? If so, show the value of such a depth limit.

Exercise 22

The graph below represents the state space of a particular problem. The nodes are the states of the problem, the edges connect a state with its successors, and the values in the edges represent the cost of the operators to move from one state to the corresponding successor. The initial state of the problem is the node A, and the final states are the nodes H and I. The heuristic estimates of the nodes are shown in the table.



Nodo	$h(n)$
A	7
B	7
C	2
D	7
E	4
F	9
G	1
H	0
I	0

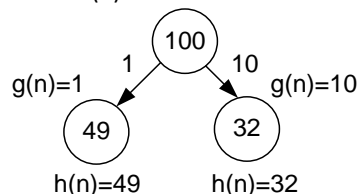
Apply an **uniform cost search** process and an **A algorithm** over the graph. In both cases, show the lists of OPEN and CLOSED nodes, the obtained solution and the cost of such a solution. Avoid repeated states. Do both strategies find the optimal solution? Is the function $h(n)$ admissible? Why?

Exercise 23

Consider a search space where the initial node is labelled with the number 100 ($n=100$), and expanding any node 'n' always results in two successor nodes:

- 1) one of the successors is a node labelled as $\text{int}(n/2)-1$, and the cost of applying this operator is 1,
- 2) the other successor is a node labelled as $\text{int}(n/3)-1$, and the cost of applying this operator is 10

Let's assume a heuristic function ' $h(n)$ ' such that $h(n)=n$. The tree below shows the first level of the tree.



Taking into account that a goal state in this problem is any node 'n' such that $n < 0$ (negative labelled node):

- Draw the search tree resulting from applying an A algorithm ($f(n)=g(n)+h(n)$). Show clearly the g-values and h-values of each node. Show the solution path and its cost.
- Draw the search tree resulting from applying a greedy best-first search. Show the solution path and its cost.
- Is the A-algorithm an A* search? Why? Justify your answer.

Exercise 24

We have five coins placed in a row as indicated in figure (a). The coin head is represented by **H** and the tail by **T**.

The only legal movement in this problem is to turn over (from **H** to **T** or from **T** to **H**) two contiguous (adjacent) coins; that is, turn over coin1-coin2 (coin at position 1 and coin at position 2), coin2-coin3, coin3-coin4 or coin4-coin5. This operation costs 1. We wish to get the situation shown in figure (b).

1	2	3	4	5
H	T	H	T	H

(a) Initial state

1	2	3	4	5
T	T	T	H	T

(b) Final state

Heuristic function. The heuristic function $h(n)$ checks the coins in the row in two steps, first step from left to right, and second step from right to left. At each step, the contiguous coins that are analyzed are:

Step1 (from left to right): check coin 1 with coin2, and coin 3 with coin 4

Step2 (from right to left): check coin 5 with coin4, and coin 3 with coin 2

h-values for pairs of contiguous coins are as follows:

$h(\text{coin X in its right place, coin Y in its right place}) = 0$ points

$h(\text{coin X out of place, coin Y out of place}) = 1$ point

$h(\text{coin X in its right place, coin Y out of place}) = 2$ points

$h(\text{coin X out of place, coin Y in its right place}) = 2$ points

Thus, if two contiguous coins are at their right place then the h-value for the pair of coins is 0. If the two contiguous coins are both out of place then the h-value for the pair is 1. If one of the two coins is out of place and the other is in its right place then the h-value for the pair of coins is 2.

The $h(n)$ function for a state 'n' of the problem is $h(n) = \min(h_left_right(n), h_right_left(n))$ (minimum value between the h-value of step1 and step2) where:

$$h_left_right(n) = h(\text{coin1,coin2}) + h(\text{coin3,coin4})$$

$$h_right_left(n) = h(\text{coin5,coin4}) + h(\text{coin3,coin2})$$

For instance, for the initial configuration in figure (a), the value of $h(n)$ is:

$$h(\text{HTHTH}) = \min(2+1, 1+2) = 3$$

Answer the following questions by providing **APPROPRIATE JUSTIFICATIONS** to all the questions:

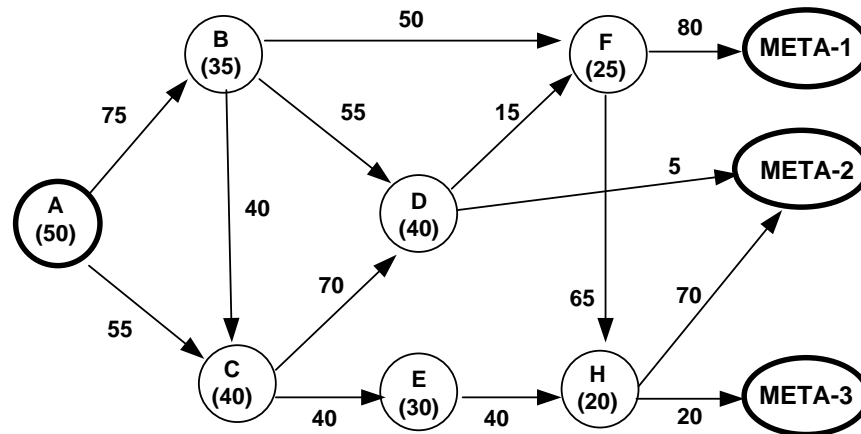
- 1) Draw the search tree resulting of applying an A-algorithm ($f(n) = g(n) + h(n)$) over the initial state in figure (a). Show clearly the f-value of each node, the node expansion order, and the OPEN list at each iteration. Which solution, and with which cost, is found with this algorithm? Is it an optimal solution?
- 2) The heuristic function 'h', is it an A* heuristic? Why? Justify your answer.
- 3) Assuming the user selects $m=5$ as maximum depth level for a depth-first search, which solution will this algorithm return and which is the cost of this solution?

IMPORTANT NOTES for the node expansion:

- a. Note 1: Apply movements in this order to create the search trees: first, coin1-coin2, then coin2-coin3, then coin3-coin4 and finally coin4-coin5.
- b. Note 2: If two nodes have the same f-value, expand first the **deepest** node. If both are at the same level, expand first the **oldest** node.
- c. Note 3: Avoid **ALWAYS** repeated nodes.

Exercise 25

The graph shown below represents a search space where the initial state is node A, the goal states are META-1, META-2 and META-3, and edges denote the accessible states. The values on the edges represent the cost of the arc to go from one node to the other, and the numbers inside the nodes denote the value of a heuristic function $h(n)$.



a) Draw the search tree for each of the three following search strategies. Show clearly the f -value of each node and the node expansion order. Show also the solution path found, the reached goal state and the cost of the solution.

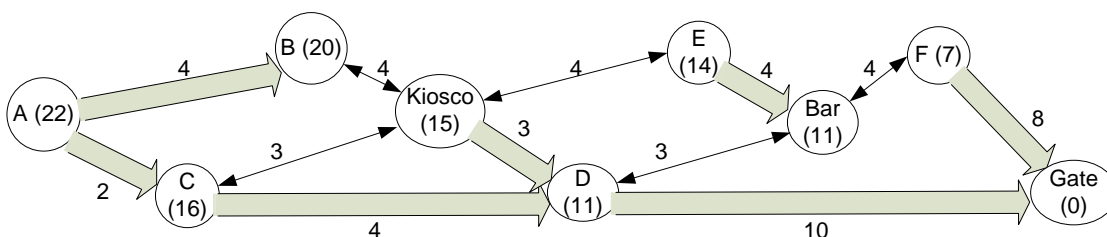
- a.1) An A-algorithm ($f(n)=g(n)+h(n)$)
- a.2) A greedy best-first algorithm
- a.3) A uniform cost strategy

Note: If two nodes have the same f -value, expand in alphabetical order

b) Does any of the methods in points a) and b) return the optimal solution? Is $h(n)$ an A* heuristic function? Why? Justify your answer.

Exercise 26

A passenger in an airport is located at spot A and wants to reach the gate (Gate) passing by the kiosk (Kiosco) to buy the newspaper and the bar (Bar) to have a coffee (see figure below). The thick lines represent passenger conveyors that run in only one direction whereas the thin lines represent corridors to move from one spot to another in both directions. Numbers labelling the edges (conveyors or corridors) denote the time to move from spot X to spot Y. Values inside the nodes show the estimated time to move from spot X to the Gate.



- a) Calculate the solution path from spot **A** to **Gate** by applying an A algorithm and avoiding repeated states. A state in this problem includes the following information: the spot at which the passenger is, whether the passenger has already dropped by the **Kiosko** or not, and whether the passenger has already passed by the **Bar** or not. Consequently, two states are repeated if they contain exactly the same information regarding these three items, no matter the path to reach the states. For instance, the path $A \rightarrow B \rightarrow \text{Kiosco} \rightarrow D$ and the path $A \rightarrow C \rightarrow \text{Kiosco} \rightarrow D$ reach the same state as in both states the passenger is at spot D and the passenger has already passed by the Kiosco but not yet by the Bar. Put another way, a solution path is not a path that simply reaches the Gate but a path that also traverses the Kiosco and the Bar.

Show clearly:

- The tree that results from the search process. Show the solution path.
- The order of the nodes expansion.
- The cost of the solution path.
- The number of generated and expanded nodes.

Note: If two nodes have the same f -value, expand first the deepest node.

- b) The heuristic function 'h', is it an A* heuristic? Why? Justify your answer.
- c) Apply again an A algorithm with the following heuristic function: given a state 's', where the current location of the passenger is 'n', the estimated cost to reach the Gate is: the $h(n)$ value indicated in the figure above, plus 10 if the passenger has not passed by the Kiosco yet, plus 10 if the passenger has not passed by the Bar yet. For example, given the state 's' resulting from the path $A \rightarrow B \rightarrow \text{Kiosco} \rightarrow D$, $h(s)=11$ (because the passenger is at spot D) + 10 (because the passenger has not visited the Bar yet) = 21.

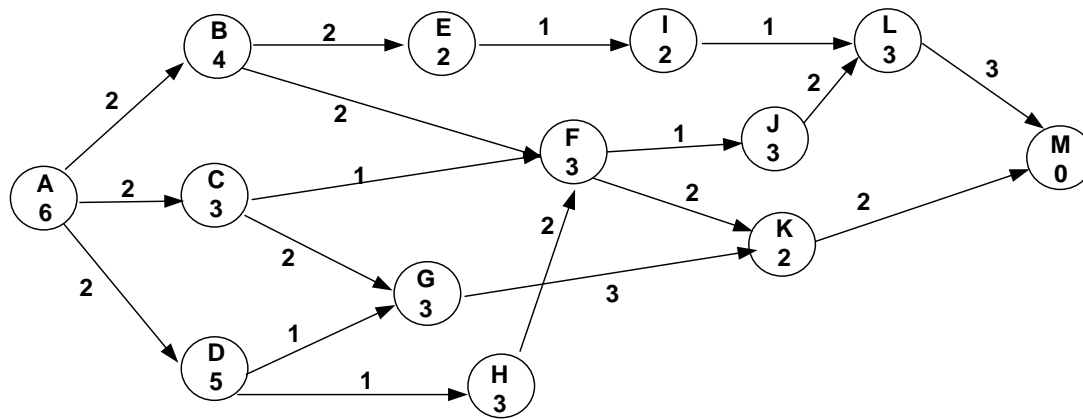
Show clearly:

- The tree that results from the search process. Show the solution path, the order of expanding the nodes, the cost of the solution path, and the number of generated and expanded nodes.
- Compare this heuristic with the one in section a). Is this new heuristic admissible? Justify your answer.

Exercise 27

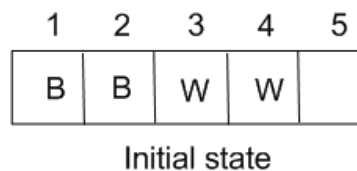
The graph below represents a search space where the initial state is node A and the goal state is M. The values on the edges represent the cost of the arc to go from one node to the other, and the numbers inside the nodes denote the value of a heuristic function $h(n)$.

- a) Apply an A algorithm to the graph of the figure (avoid repeated states). Draw the search tree, indicate clearly the f -values of the nodes and the order in which nodes are expanded. Say the cost of the solution found, the number of generated and expanded nodes. **NOTE:** Note: If two nodes have the same f -value, expand in alphabetical order.
- b) The solution found by the A algorithm, is it optimal? The heuristic function 'h', is it an A* heuristic? Why? Justify your answers.
- c) The 'h' function is non-monotone (no consistent). Say the values in the graph and/or the tree that indicate the function 'h' is non-monotone and explain the consequences of the absence of this property in the developed tree.
- d) Draw the search tree that results from the application of a depth-first strategy with maximum depth level $m=4$ (avoid repeated states). Show the OPEN and CLOSED lists in each iteration. Which solution returns DFS? Which is the cost of this solution? **NOTE:** Note: If two nodes are at the same depth level of the tree, expand in alphabetical order.



Exercise 28

Given the following configuration of a lineal puzzle:



where there are two black pieces (B), two white pieces (W) and one empty position, and considering the following actions:

- A piece can move to an empty adjacent position with cost 1.
- A piece can jump over another one (only one) to be placed in the following empty position with cost 1. If the piece that jumps and the jumped one have different colours, the jumped one changes its colour.

The objective is to have all the pieces of the board **black coloured** in any position. Thus, the end position of the empty square is indifferent.

- Consider the following heuristic function: $h(n) = \sum_{i=1}^5 b(i) * c(i)$, where:

- $b(i)=1$, if square (i) contains a W piece; $b(i)=0$, otherwise
- $c(i)=1$, if the empty square is adjacent to square (i), i.e., if the empty space is in square (i-1) or square (i+1); $c(i)=2$, otherwise.

In short, the heuristic only considers the W pieces. For each W piece, it counts 1 if the empty square is adjacent to the W piece or it counts 2 if the empty square is not adjacent to the W piece.

- Show the search tree generated by applying an A algorithm with $f(n)=g(n)+h(n)$.
- Show the *order of expansion* of the nodes in the tree
- Show the solution found and the cost of the solution

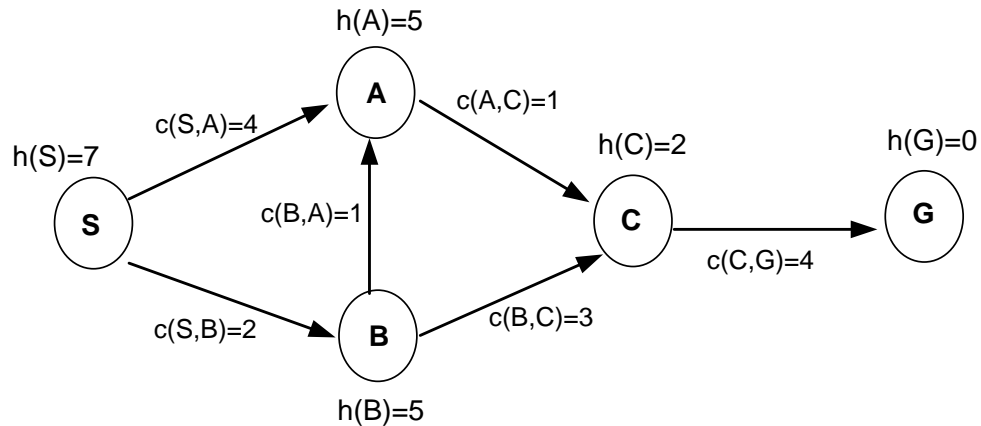
NOTES: (a) Avoid repeated states. (b) If two nodes have the same f-value, expand first the **deepest node**.

- The 'h' function described in 1), is it admissible? Justify your answer.
- Without developing a new search tree, answer the following questions:
 - How many iterations (different search trees) would Iterative Deepening do to find a solution? Justify your answer.

- b) If the cost of jumping over another piece were 2 instead of 1, would the above A algorithm find the same solution? Justify your answer.

Exercise 29

Given the following state space where S is the initial state and G is the goal state:

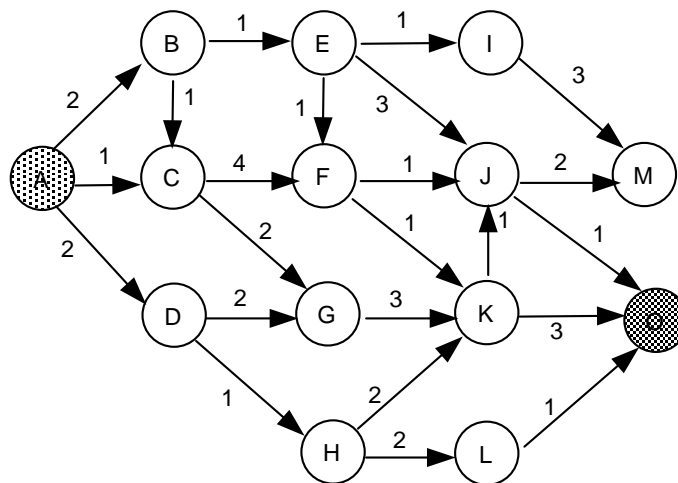


, $h(X)$ denotes the value of applying the heuristic function 'h' in state X; $c(X,Y)$ is the cost of the operator to move from state X to state Y. Answer the following questions:

- Apply an A algorithm with $f(n)=g(n)+h(n)$. Does it find the optimal solution?
- Is the 'h' function admissible? Is the 'h' function consistent?

Exercise 30 (Exam 2013)

Arcs in the graph below are labeled with the cost to go from one node to the other. The table shows the estimated cost 'h' to reach the solution. The initial state is the node 'A' and the final state is the node 'O'.

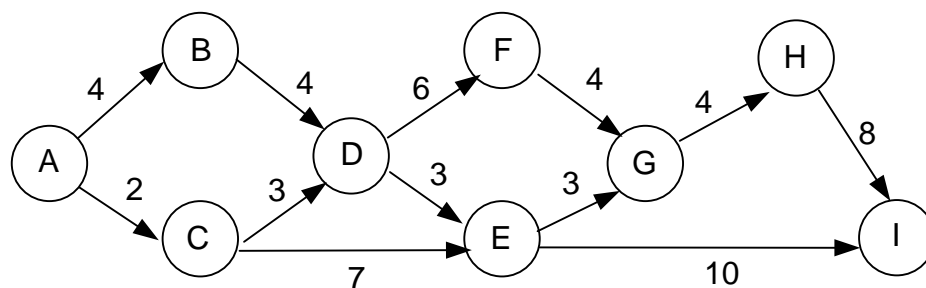


n	A	B	C	D	E	F	G	H	I	J	K	L	M	O
h(n)	6	5	6	6	3	5	5	4	8	2	2	1	5	0

1. Show the search tree that would result from the application of an A algorithm ($f(n)=g(n)+h(n)$). Apply the graph-search version avoiding repeated states. Indicate the number of generated nodes and expanded nodes. Show clearly the value of the evaluation function ($f(n)$) in each node and the order of expansion of the nodes. If two nodes have the same f -value, expand first the node that comes alphabetically before.
2. According to the problem data and the tree developed in the previous section, does the algorithm return the optimal solution? Is the heuristic function admissible? And consistent (monotone)? Provide justifications for all your answers.
3. Answer the following questions without need of building a tree. Justify all your answers.
 - a. Which strategy would you use if we want to find a solution that goes through the least number of nodes? Show a solution that this strategy would find.
 - b. If we apply an iterative deepening depth-first algorithm, in which level would it find a solution? Why?
 - c. If we apply a depth-first algorithm and the user does not set a maximum depth level, would the algorithm find a solution? Why?

Exercise 31 (Exam 2013)

Consider the graph below where arcs are labeled with the cost to go from one node to the other. The table shows the estimated cost (h) to reach a solution. The initial state is node 'A' and the goal state is node 'I'.



n	A	B	C	D	E	F	G	H
$h(n)$	15	14	13	12	11	10	9	8

- 1) Assuming we apply a breadth-first algorithm (in case of the same value of the function ' $f(n)$ ', the node that comes alphabetically before is expanded first) with control of repeated states (discard deeper nodes or nodes that have been expanded before in case of two nodes at the same depth level).
 - a) Write the nodes of the solution path from node A to node I.
 - b) How many nodes have been generated and how many expanded in the breadth-first tree?
- 2) Assuming we apply the graph version of an A algorithm with control of repeated states:
 - a) Write the nodes of the solution path from node A to node I and the cost of the solution path. Is the solution found optimal?
 - b) How many nodes have been generated and how many expanded in the developed tree?
 - c) Show the nodes in the order in which they are expanded.

3) Answer briefly the following questions. Justify your responses:

- a) Is the heuristic function admissible? Why?
- b) Is the heuristic function consistent? Why?
- c) If we apply an iterative deepening (ID) algorithm for this problem, which solution would ID find? Show the nodes of the solution path as well as the total number of generated nodes.