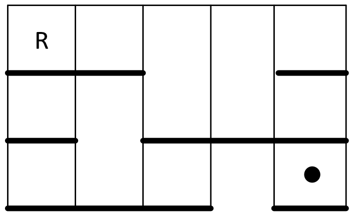
Intelligent Systems – Exam Block 1, November 13, 2020
Test (1.75 puntos) score: max (0, (#correct_answers – #errors/3)*1.75/9)

Last name(s):								**Name**:
Group:	A	B	C	D	E	F	G	

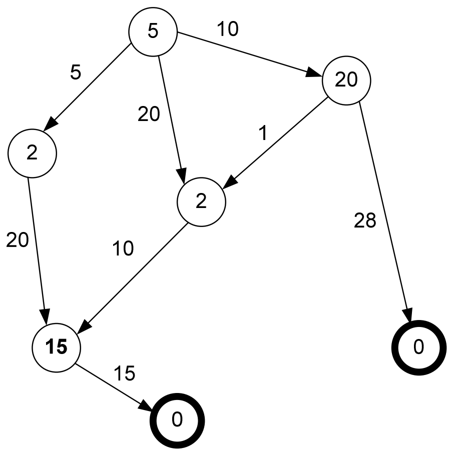
1) The figure below shows a grid with a robot in position (x,y)=(1,3) which wants to reach the square marked with a spot in position (x,y)=(5,1). There is no floor in some of the squares (holes), and the robot can never be located in such squares. The robot can move horizontally (right and left), vertically (up and down) and diagonally. For example, from position (1,3) the robot can only move right and down but not diagonally. The cost of each operator is 1. Show the **CORRECT** answer:



A. The cost of the optimal solution for this problem is 6.
B. The heuristic "Manhattan distance" applied to this problem is an admissible heuristic
C. The expansion of node (x,y)=(5,2) generates 4 children nodes.
D. None of the above answers is correct.

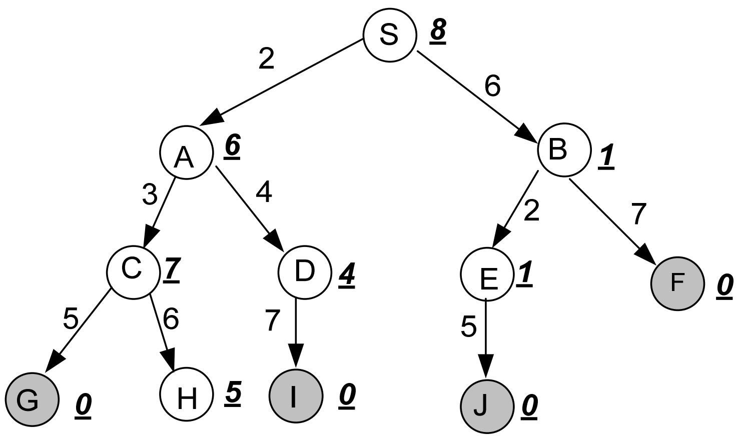
---

2) Given the search space of the figure, where the values on the edges are the cost of operators, the heuristic estimate of each node is the value inside it, and the initial node is the one with h(n)=5, show the **CORRECT** answer:



A. h(n) is not admissible.
B. The TREE-SEARCH version of an A algorithm with control of repeated states in OPEN does not return the optimal solution.
C. The GRAPH-SEARCH version of an A algorithm with control of repeated nodes in OPEN and without re-expansion of repeated nodes in the CLOSED list does not return the optimal solution.
D. A BREADTH-first search returns the optimal solution.

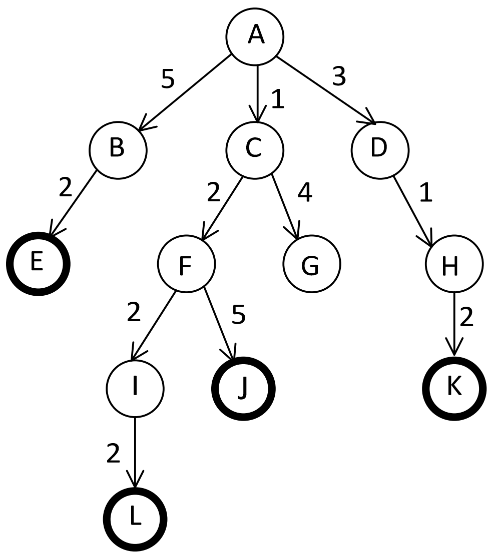
3) Given the search space of the figure, where the values on the edges are the cost of the operators and the heuristic estimate is the underlined value off the node, show the **CORRECT** answer:



A. A DEPTH-first search will generate the same or more nodes than a UNIFORM COST search.
B. The function h(n) is admissible.
C. An algorithm of type A returns the optimal solution.
D. None of the above answers is correct.

---

4) The figure below shows the space generated with a search algorithm. The bold double-circled nodes are goal states, and the algorithm finds **node K** as the first solution. The values on the edges are the cost of the operators. Which uninformed search algorithm has been applied?



A. BREADTH-first search.
B. UNIFORM COST search.
C. DEPTH-first search with maximum depth level m=3.
D. ITERATIVE DEEPENING search.

5) Let be the search tree of the above question (question 4), and a heuristic function h(n) that we know is admissible. Show the **FALSE** statement:

A. h(J)=h(E)
B. g(L) <= g(D)+h(D)
C. h(C) <= 6
D. h*(A)=6

---

6) Given the initial working memory WM={(lista e b a c d e a) (lista1 a c a d e e f g)} and the rule R1, show the state of the final WM.

```
(defrule R1
  ?f <- (lista  $?x ?a $?y)
        (lista1  $?p ?a $?q ?a  $?r)
=>
      (retract ?f)
      (assert (lista $?x $?y)))
```

A. {(lista e b a c d f g) (lista1 a c a d e e f g)}
B. {(lista b) (lista1 a c a d e e f g) }
C. {(lista b c d) (lista1 a c a d e e f g)}
D. {(lista e b a c d) (lista1)}

---

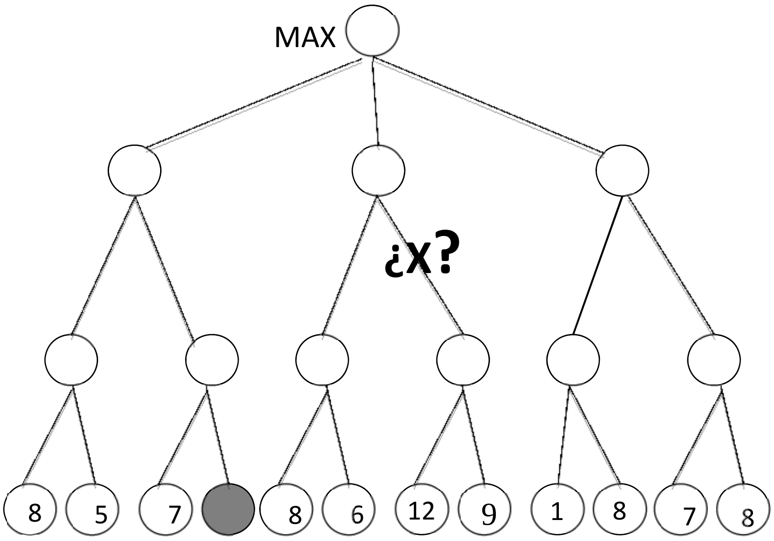
7) Indicate the **CORRECT** final result when we execute the following RBS with initial WM={(lista 34 77 34)}:

```
(defrule R1
  (declare (salience 15))
    ?f <- (lista $?x1 ?num $?x2)
=>
   (retract ?f)
   (printout t "Message 1" crlf))

(defrule R2
  (declare (salience 20))
    ?f <- (lista ?num $?x ?num)
=>
   (retract ?f)
   (printout t "Message 2" crlf))

(defrule R3
  (declare (salience 30))
=>
  (printout t "Message 3"))
```

A. "Message 3" is shown three times.
B. "Message 2" is shown once and then "Message 3" is shown once.
C. "Message 3" is shown once and then "Message 2" is shown once.
D. "Message 3" is shown once, then "Message 2" is shown once and then "Message 1" is shown once.

---

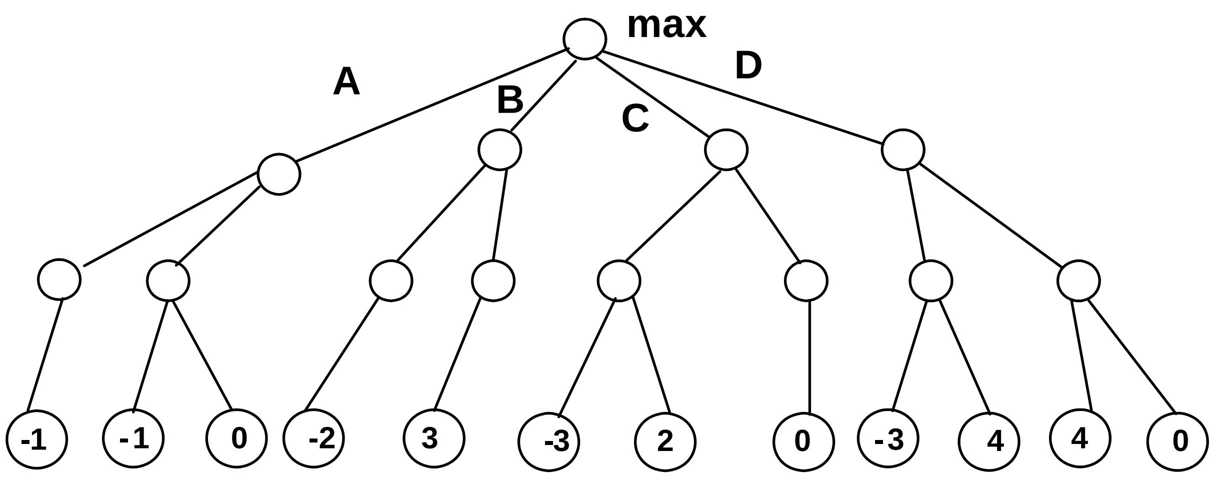
8) Given the game search of the figure below, and assuming we apply an alpha-beta procedure, show the value that the shadowed node should have in order to get the cut-off shown in the figure.



A. Any value of the shadowed node would produce the cut-off.
B. A value lower than 8.
C. A value equal or higher than 8.
D. The cut-off can never happen (or none of the above answers is correct).

---

9) Given the game search space of the figure, if we apply an alpha-beta procedure, how many nodes are not needed to be generated compared to the application of a MINIMAX algorithm?

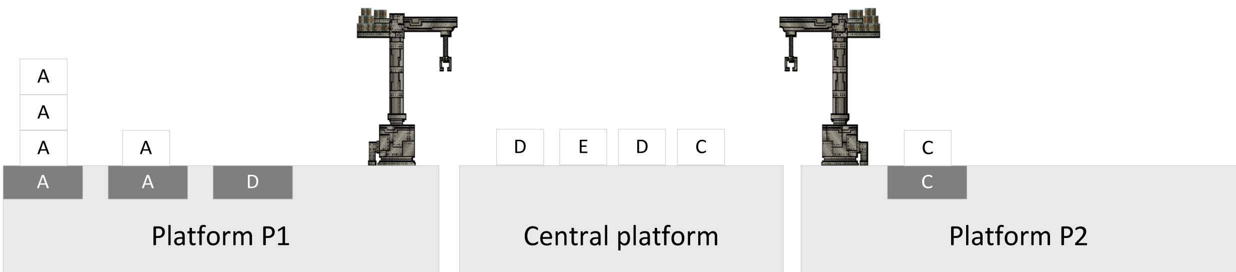


A. 3
B. 4
C. 5
D. 6

Intelligent Systems – Exam Block 1, November 13, 2020
Problem: 2 points

The port of a city has a container handling system that works as follows:

1) There is a central platform, and two platforms named P1 and P2. Platforms P1 and P2 each has a crane (see the figure below).
2) A number of containers are in the central platform. All containers are accessible, that is, they are all on the floor. Containers belong to a class (A, B, C, D, …).
3) Every container is associated to a single type, either P1 or P2, which indicates its destination platform. The type Pi of each container is determined by the class of the container. That is, containers of a class X are meant to go to a particular platform Pi.
4) The crane of a platform Pi can only pick up one container at a time from the central platform whose destination is Pi.
5) The crane of a platform Pi drops a grabbed container of class X in an existing heap of the same class X in Pi. Heaps cannot hold more than 3 containers. In the example of the figure, two heaps of class A and one heap of class D stand in platform P1; one of the heaps of class A has already 3 containers, the other heap of class A has only one container and the heap of class D is empty.



The structure of the dynamic pattern is the following (you are not allowed to change the structure of this pattern):

(port b^m platform T1 [heap c^s n^s]^m crane T1 p^s platform T2 [heap c^s n^s]^m crane T2 p^s)

where: $\mathbf{b}^m \in$ {A, B, C, D, …} each element in $\mathbf{b}^m$ denotes a container of the corresponding class in the central platform

$c^s \in$ {A, B, C, D, ...} class of the heap, it denotes the class of the containers the heap can stack

$n^s \in$ [0 3] number of containers in the heap

$p^s \in$ {empty, A, B, C, …} class of the container grabbed by the crane or empty otherwise

NOTE: For a crane to drop a grabbed container in a heap, a heap of the same class as the container must exist in the platform. The heap can be empty (0 containers). An ***empty heap*** and a ***non-existing heap*** is not the same. In the latter case, the heap does not appear in the pattern.

Initially, there are no heaps in any of the two platforms. An empty heap will be created with a CLIPS rule.

Answer the following questions using CLIPS language and assuming a GRAPH-SEARCH expansion:

a) (0.3 points) Write the **initial Working Memory** taking into account that the central platform contains the following containers: A E B B C D B B A E D. We know that containers of class A and D go to platform P1, and containers of class B, C and E go to platform P2. Neither of the two platforms contains a heap, not even an empty heap, and the cranes are empty. For the static facts, pick the pattern you consider to be the most appropriate.

b) (0.5 points) Write only one rule "initial_heap" that creates an empty heap in a platform Pi under certain conditions. Specifically, if there is a container of class X in the central platform, there is no heap of class X in its destination platform Pi and the crane is empty, the rule will create an empty heap in Pi. An empty heap is represented in the figure with a dark grey rectangle.

c) (0.7 points) Write only one rule "pick_up_container" that allows an empty crane of a platform Pi to pick up a container from the central platform whose destination is Pi. The rule must satisfy the condition that a heap of the same class as the container already exists in Pi and the heap is not full (i.e., it has less than three containers).

d) (0.5 points) Write a "stop" rule. This rule will trigger when all the containers are in their corresponding platforms. The rule will display the total number of created heaps.

```
(deffacts datos
      (puerto A E B B C D B B A E D plataforma T1 grua T1 vacio plataforma T2 grua
T2 vacio)
      (contenedor T1 A D)
      (contenedor T2 B C E))


(defrule initial_heap
      (puerto $?p1 ?p $?p2 plataforma ?t $?l grua ?t vacio $?r)
      (test (not (member$ ?p $?l)))
      (contenedor ?t $? ?p $?)
      =>
      (assert (puerto $?p1 ?p $?p2 plataforma ?t $?l pila ?p 0 grua ?t vacio $?r)))


(defrule pick_up_container
      (puerto $?p1 ?p $?p2 plataforma ?t $?e1 pila ?p ?c $?e2 grua ?t vacio $?r)
      (contenedor ?t $? ?p $?)
      (test (< ?c 3))
      =>
      (assert (puerto $?p1 $?p2 plataforma ?t $?e1 pila ?p ?c $?e2 grua ?t ?p $?r)))


(defrule stop
      (declare (salience 100))
      (puerto plataforma T1 $?p1 grua ? vacio  plataforma T2 $?p2 grua ? vacio )
      =>
      (printout t "el número de pilas es " (+ (div (length$ $?p1) 3)(div (length$
$?p2) 3)) crlf)
      (halt))
```