Intelligent Systems – Exam Block 1, October 31, 2019
Test A (1.75 points) score: max (0, (#correct_answers – #errors/3)*1.75/9)

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

1) Let be the following pattern for the Hanoi towers problem:

(hanoi [tower tw^s $d1^s$ $d2^s$ $d3^s$ $d4^s$]^m) ;; $tw^s \in$ {T1,T2,T3} $d_i^s \in$ [0-4]

, for which two examples of facts are (hanoi tower T1 2 4 0 0 tower T2 0 0 0 0 tower T3 1 3 0 0) (hanoi tower T1 2 4 0 0 tower T2 1 3 0 0 tower T3 0 0 0 0).

Is the following rule correct for moving a disk from tower T3 to any of the other two towers T1, T2 on its left?

```
(defrule move-disk-from-T3-to-Tx
  (hanoi  $?rest1 tower ?Tx  ?d2  $?rest2 0  $?rest3 tower T3 ?d1 ?a ?b ?c)
  (test (or (= ?d2 0) (< ?d1 ?d2)))
 =>
 (assert (hanoi $?rest1 tower  ?Tx   ?d1 ?d2  $?rest2  $?rest3 tower T3 ?a ?b ?c 0)))
```

A. Yes, the rule is correct.
B. The rule would be correct if we check (not (member tower $?rest2)) in the LHS of the rule.
C. Besides the check indicated in response (B), it would be also necessary to check (<> ?d1 0) in the LHS of the rule.
D. Besides the two checks shown in (B) and (C), it would be also necessary to check (> (length $?rest2) 0) in the LHS of the rule.

---

2) Let be a RBS with initial Working Memory IniWM={(lista 3 4 5 6 6 6 8 9)}, and the following two rules:

```
(defrule R1
  ?f <- (lista $?x ?z ?y $?w)
        (test (= ?z ?y))
 =>
 (assert (lista $?x ?z ?y $?w)))
```

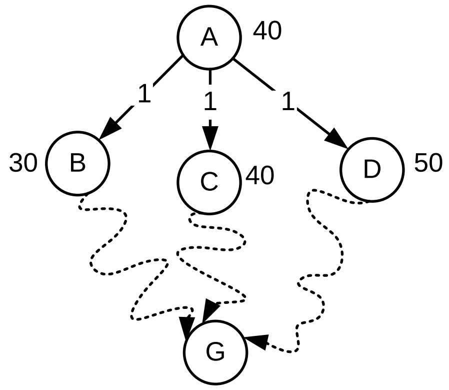
```
(defrule R2
  ?f <- (lista $?x ?z ?y $?w)
        (test (> ?z ?y))
 =>
 (assert (lista $?x ?z ?y $?w)))
```

What will be the contents of the Agenda (Conflict Set) after the first pattern-matching?

A. Two instances of rule R1 and two instances of rule R2.
B. Two instances of rule R1.
C. Two instances of rule R1 and one instance of rule R2.
D. No instance will be inserted in the Agenda.

---

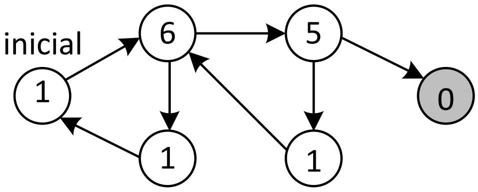
3) We want to find a solution path from the initial state A to the single goal state G in the search space of the figure below. Arcs have a unitary cost and the graph shows the heuristic value of each node, h(n), which we know is an admissible heuristic function. Assuming we apply an algorithm of type A (f(n)=g(n)+h(n)), with re-expansion of nodes, it is **TRUE** that:



A. If we get a solution path from node A to node G whose cost is 40, this will be an optimal solution.
B. If we get a solution path from node A to node G whose cost is 51, this will not be the optimal solution.
C. The application of the algorithm A will not return the optimal solution.
D. From the data displayed in the graph, we can say h(n) is consistent.

---

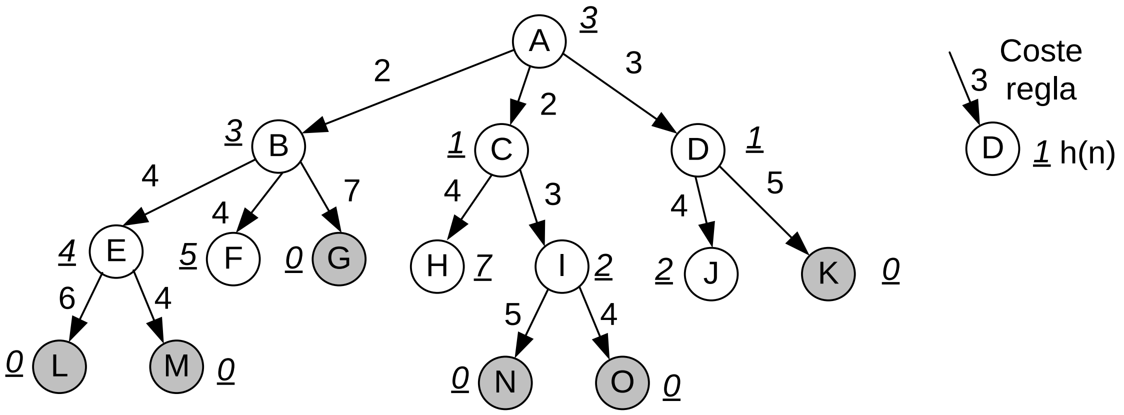
4) Let's assume we apply an algorithm of type A (f(n)=g(n)+h(n)) to the state space shown in the figure below. The shadowed node is the goal state, all arcs have a cost of 1, and the value of the heuristic function h(n) is shown within each node. Mark the CORRECT answer:



A. The algorithm cannot be applied to the state space because h(n) is not a consistent heuristic function.
B. The application of the algorithm will get stuck in an endless loop and it will not end.
C. The application of the algorithm will find the solution, but it does not necessarily have to be the optimal one.
D. The application of the algorithm will find the optimal solution.

---

5) In the state space below, the shadowed nodes are goal states. Assuming we apply a search of type A ($f(n)=g(n)+h(n)$) to find the solution, how many nodes, including the node A, will be generated?



A. 10.
B. 11.
C. 13.
D. 15.

---

6) Given the search space of question 5, and assuming we randomly select a node when two or more nodes have the same value of $f(n)$, show the **CORRECT** answer:

A. The solution found by a Depth First Search with m=3 (maximum depth level) will be any of the shadowed nodes.
B. The solution found by a Uniform Cost strategy will be the node G or the node N.
C. The solution found by an Iterative Deepening strategy will be one of the shadowed nodes among L, M, N or O.
D. None of the above answers is correct.

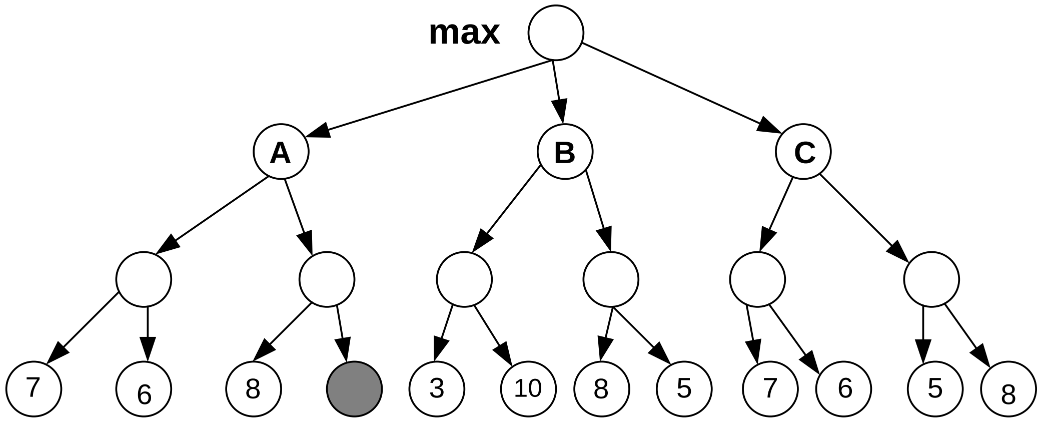
---

7) Let be a 4x4 grid where a robot is located at the bottom left cell, (x,y)=(1,1), and it aims at reaching the top right cell, (x,y)=(4,4). The robot can move horizontally, vertically or diagonally, and all moves have a cost of 1. Mark the **INCORRECT** answer:

A. The cost of the optimal solution to this problem is 3.
B. The heuristic Manhattan distance applied to this problem is an admissible heuristic.
C. The application of a Breadth First Search strategy will return the optimal solution.
D. The expansion of the node (x,y)=(2,3) will yield 8 child nodes.

---

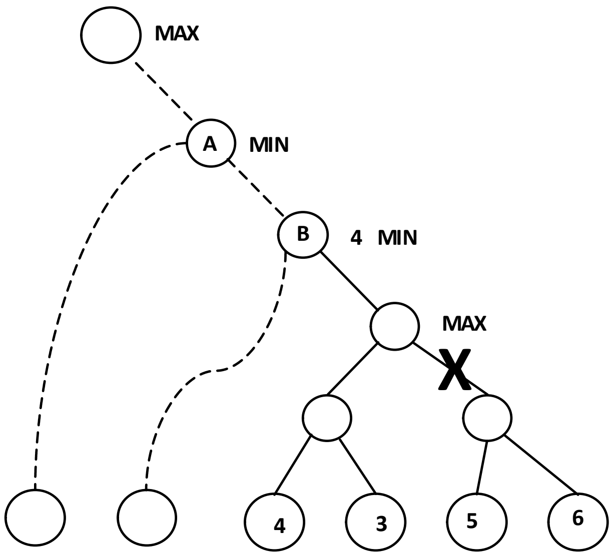
8) Given the game search tree below, and assuming we apply an alpha-beta procedure, show the **CORRECT** answer:



A. If the shadowed node takes on a value <=8, an alpha cut-off will be produced in both node B and node C.
B. If the shadowed node takes on a value >=10, an alpha cut-off will be produced in both node B and node C.
C. Regardless the value of the shadowed node, an alpha-beta cut-off will be always produced in node B.
D. Regardless the value of the shadowed node, an alpha-beta cut-off will be always produced in node C.

---

9) In the partial alpha-beta search of the figure below, the node B has a provisional backed-up value of 4. Which provisional value should node A have so that the effective cut-off shown in the figure is produced?



A. The cutoff of the figure can never be produced.
B. A value lower or equal than 3.
C. A value higher or equal than 3.
D. A value lower than 3.

Intelligent Systems – Exam Block 1, October 31, 2019
Problem: 2 points

A factory manufactures three different ranges (categories) of a given product: mid range, high range and supreme range. Making a product of each range requires a different number of pieces of two types, A and B:

- A mid-range product is made with 3 pieces of type A and 1 piece of type B
- A high-range product is made with 2 pieces of type A and 2 pieces of type B
- A supreme-range product is made with 3 pieces of type B

The factory keeps pieces of type A and type B in stock. We also know the factory has a number of product orders of each range. The pattern that represents the dynamic information of this problem is:

(factory [piece type^s num_p^s]^m [product-order range^s num_o^s]^m)

type $\in$ {A,B}
num_p $\in$ INTEGER ;; number of pieces of the corresponding type
range $\in$ {mid, high, supreme}
num_o $\in$ INTEGER ;; number of product orders of the corresponding range

An example of initial situation is:

- The factory has 10 pieces of type A and 15 pieces of type B in stock
- The product orders are: 2 mid-range products, 1 high-range product, 1 supreme-range product

1) (0.4 points) Write the facts of the Working Memory (WM) that represents the situation given above. Specify also the patterns that you need to represent the static information of the problem and include in the WM the facts associated to these patterns. NOTE: we recommend keeping the same order of the two types of pieces (A and B) in the dynamic pattern and in the necessary static patterns.
2) (0.5 points) Write a rule to deliver one product of any range. The effects of delivering a product are (a) decrementing the number of available pieces in the factory, and (b) decrementing the number of product orders of the corresponding range.
3) (0.5 points) Write a rule to deliver all the product orders of one same range.
4) (0.6 points) Write a rule that determines, for two product orders of different range, if there are only pieces for delivering one of them. The rule must display a message of this type: "The factory has only pieces in stock for delivering one product of range XXX or of range YYY".

1) Static pattern: (range g^s A n1^s B n2^s) $g \in$ {mid, high, supreme},   n1,n2 $\in$ INTEGER

```
(deffacts datos
     (factory piece A 10 piece B 15 product-order mid 2 product-order high 1
product-order supreme 1)
     (range mid A 3 B 1)
     (range high A 2 B 2)
     (range supreme A 0 B 3))
```

2)

```
(defrule producir_uno
     (range ?range A ?pa B ?pb)
     (factory piece A ?na piece B ?nb $?x product-order ?range ?numped $?y)
     (test (>= ?numped 1))
     (test (and (>= ?na ?pa)(>= ?nb ?pb)))
     =>
     ;;(printout t "range " ?range "pieces A " ?pa " " ?na crlf)
     (assert (factory piece A (- ?na ?pa) piece B (- ?nb ?pb) $?x product-
order ?range (- ?numped 1) $?y)))
```

3)

```
(defrule producir_todos
     (range ?range A ?pa B ?pb)
     (factory piece A ?na piece B ?nb $?x product-order ?range ?numped $?y)
     (test (>= ?numped 1))
     (test (and (>= ?na (* ?pa ?numped))(>= ?nb (* ?pb ?numped))))
     =>
     (assert (factory piece A (- ?na (* ?pa ?numped)) piece B (- ?nb (* ?pb
?numped)) $?x product-order ?range 0 $?y)))
```

4)

Assuming there are enough pieces to deliver one of the two products:

```
(defrule servir_solo_uno
    (factory piece A ?na piece B ?nb $?x product-order ?range1 ?numped1 $?y
product-order ?range2 ?numped2 $?z)
     (test (and (>= ?numped1 1)(>= ?numped2 1)))
     (range ?range1 A ?pa1 B ?pb1)
     (range ?range2 A ?pa2 B ?pb2)
     (test (or (< ?na (+ ?pa1 ?pa2)) (< ?nb (+ ?pb1 ?pb2))))
     =>
     (printout t "The factory has only pieces in stock for delivering one
product of range " ?range1 " o of range " ?range2 crlf))
```

Checking there are enough pieces to deliver at least one of the two products:

```
(defrule servir_solo_uno
    (factory piece A ?na piece B ?nb $?x product-order ?range1 ?numped1 $?y
product-order ?range2 ?numped2 $?z)
     (test (and (>= ?numped1 1)(>= ?numped2 1)))
     (range ?range1 A ?pa1 B ?pb1)
     (range ?range2 A ?pa2 B ?pb2)
     (test (or (< ?na (+ ?pa1 ?pa2)) (< ?nb (+ ?pb1 ?pb2))))
     (test (or (and (>= ?na ?pa1) (>= ?nb ?pb1))
               (and (>= ?na ?pa2) (>= ?nb ?pb2))))
     =>
     (printout t "The factory has only pieces in stock for delivering one
product of range " ?range1 " o of range " ?range2 crlf))
```