

Examen conv. 2 curso 96-97

30 Junio 1997

1 hora 20 minutos

Apellidos y Nombre:

Problema 1: Búsquedas informadas [25 puntos] [25 minutos]

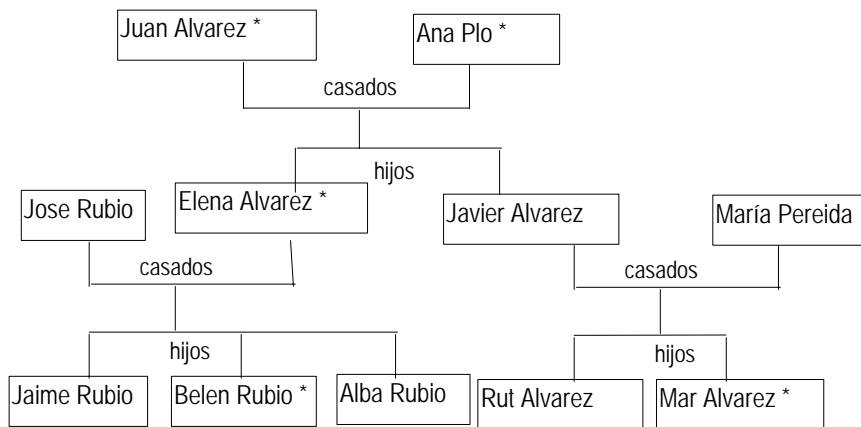
Utilizar la estrategia de búsqueda heurística A* con el fin de trazar un camino óptimo entre las ciudades de Salamanca y Teruel teniendo en cuenta el cuadro de distancias en kilómetros siguiente:

	Ávila	Zaragoza	Zamora	Teruel	Salamanca	Valladolid
Ávila		418	159		97	121
Zaragoza	418		463	181	492	367
Zamora	159	463			62	96
Teruel		181				545
Salamanca	97	492	62			115
Valladolid	121	367	96	545	115	

También es un dato conocido el siguiente cuadro de las distancias aéreas desde cada ciudad a Teruel:

	Ávila	Zaragoza	Zamora	Salamanca	Valladolid
Teruel	400	170	480	450	540

Describir de forma clara el árbol de búsqueda desarrollado, donde quede claramente establecido el estado, el operador aplicado, el desglose de informaciones utilizadas para evaluar el nodo y el número de orden en que han sido expandidos los nodos.

Problema 2: Frames/Common Lisp/Prácticas [35 puntos] [30 minutos]

Se desea utilizar el esquema de representación IERL utilizado en la práctica 4 (en un anexo del examen encontrarás parte del código de la práctica que te facilitará recordarlo) para representar la información mostrada en la figura anterior (los * indican que la persona ha fallecido).

Se pide que escribas el código necesario (tanto sus propios forms como los de sus antecesores en la jerarquía de herencia) para crear los forms de “Javier Rodriguez” y de “Eva García”.

Crea también el código necesario (a través de las ataduras procedurales de acceso adecuadas) para que todas las personas dispongan de:

- un slot denominado “descendientes_vivos” que proporcione el número de descendientes vivos que existan en ese momento,
- un slot denominado “nietos” que devuelva sus nietos,

Al tratar de escribir en los mencionados slots debe salir un mensaje por pantalla indicando que la acción no es posible.

Problema 3: Búsqueda en juegos [15 puntos] [15 minutos]

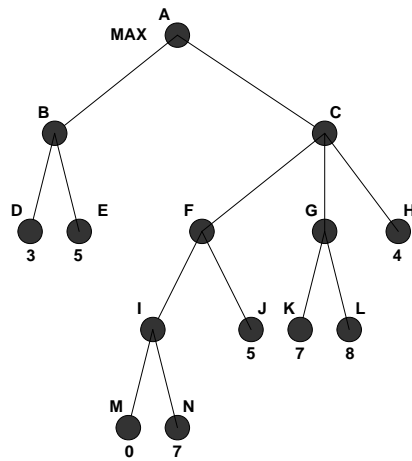


Figura A

Realiza una búsqueda mini-max en el árbol mostrado en la figura A.

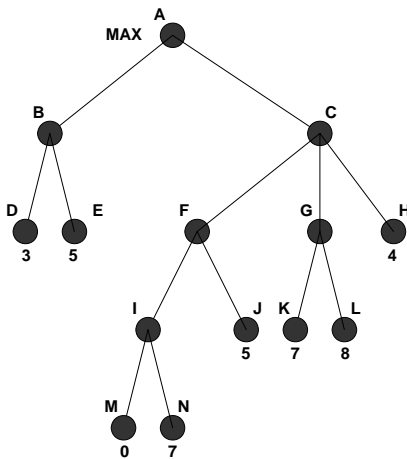


Figura B

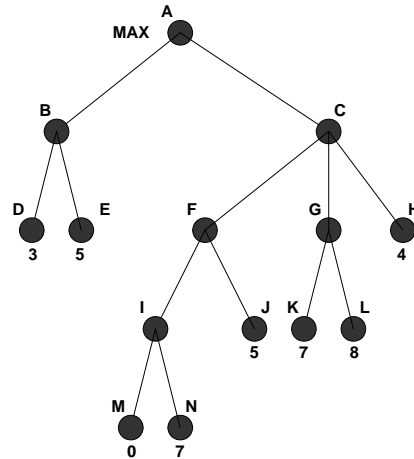


Figura C

Realiza una poda alfa-beta de izquierda a derecha en el árbol de la figura B. Realiza una poda alfa-beta de derecha a izquierda en el árbol de la figura C.

Discute porque surgen diferentes podas.

ANEXO 1

Ejemplo de utilización del esquema de representación IERL (práctica 4)

```

;;; CALCULA-RUIDO es una funcion IF-NEEDED que calcula el nivel de ruido de un apartamento.
;;; Si existe un valor = lo toma, si no lo calcula dependiendo de la altura del piso.
;;;
(defun calcula-ruido (form slot)
  (let* ((valor (find-aspect-from-supers form 'ruido-calle '=))
         piso)
    (cond (valor valor)
          (t (setq piso (get-value form 'piso))
             (cond ((> piso 15) 'muy-bajo) ((> piso 8) 'bajo) ((> piso 4) 'medio)
                   (> piso 1) 'alto) (t 'muy-alto))))))

(defun gbd ()
  (setq *tabla-de-forms* nil)
  (let ((la-form (create-form :is-a nil :name 'object))
        )
    (form :name 'cosa
          :is-a (find-form 'object))

    (form :name 'vivienda
          :is-a (find-form 'cosa))

    (form :name 'apartamento
          :is-a (find-form 'vivienda)
          :slots (list (list 'ruido-calle 'if-needed #'calcula-ruido)))

    (form :name 'chalet
          :is-a (find-form 'vivienda)
          :slots (list (list 'ruido-calle 'if-needed #'calcula-ruido-en-casa) ) )

    (form :name 'chalet-montesol-5
          :is-a (find-form 'chalet)
          :slots '((calle-nombre = urbanizacion-montesol)
                  (calle-numero = 5)
                  (numero-habitaciones = 5)
                  (pisos = 2)
                  (distancia-carretera = 200))))))

```

ANEXO 2

Alfa: (para un nodo MAX) es el valor más **alto** visto hasta el momento de los valores finales, calculados hacia atrás, de sus sucesores.

Beta: (para un nodo MIN) es el valor más **bajo** visto hasta el momento de los valores finales, calculados hacia atrás, de sus sucesores.

Poda (fin de la llamada recursiva)

Puede suspenderse la exploración por debajo de cualquier nodo MIN que tenga valores de Beta menores o iguales que el valor Alfa de cualquiera de sus nodos MAX ascendientes suyos.

Puede suspenderse la exploración por debajo de cualquier nodo MAX que tenga valores de Alfa mayores o iguales que el valor Beta de cualquiera de sus nodos MIN ascendientes suyos.