

Apellidos:

Nombre:

**Ejercicio 1** [2.5 puntos] El procedimiento de búsqueda  $A^*$  acotada es una variante del procedimiento de búsqueda  $A^*$  de forma que sólo se añaden a la lista de ABIERTOS aquellos nodos tales que la suma de su coste y su heurística es menor que  $m$  (donde  $m$  es una cota dada). Definir en Lisp el procedimiento `(busqueda-a-estrella-acotada :cota m)` de manera que, dado  $m$ , devuelva el nodo final obtenido aplicando la búsqueda  $A^*$  acotada con cota  $m$ .

---

**Ejercicio 2** [2.5 puntos] Se considera el siguiente problema

```
(defparameter *estado-inicial* (list 0 0))

(defparameter *estado-final* (list 3 2))

(defun es-estado-final (estado)
  (equal estado *estado-final*))

(defparameter *obstaculos* '((0 1) (0 2) (0 3)))

(defun es-seguro (estado)
  (not (member estado *obstaculos* :test #'equal)))

(defparameter *operadores* '(izquierda arriba derecha abajo))

(defun izquierda (estado)
  (list (- (first estado) 1) (second estado)))

(defun arriba (estado)
  (list (first estado) (+ (second estado) 1)))

(defun derecha (estado)
  (list (+ (first estado) 1) (second estado)))

(defun abajo (estado)
  (list (first estado) (- (second estado) 1)))
```

```
(defun aplica (operador estado)
  (let ((nuevo-estado (funcall (symbol-function operador)
                                estado)))
    (when (es-seguro nuevo-estado) nuevo-estado)))
```

```
(defun coste-de-aplicar-operador (estado operador) 1)
```

```
(defun heuristica (estado)
  (+ (abs (- (first estado) (first *estado-final*)))
      (abs (- (second estado) (second *estado-final*)))))
```

Explicar, indicando los estados analizados y la evolución de ABIERTOS, la solución del problema obtenida mediante

```
(busqueda-a-estrella-acotada :cota 5)
```

**Ejercicio 3** [2.5 puntos] Comentar las principales diferencias entre los algoritmos A\* y A\* acotada desde los siguientes puntos de vista:

- \* completitud
- \* complejidad en espacio

**Ejercicio 4** [2.5 puntos] Aplicar la estrategia alfa-beta al siguiente árbol

