



Examen de Teoría de la Programación

E. U. ING. TEC. EN INFORMÁTICA DE OVIEDO

Final Febrero – Curso 2008-2009

7 de febrero de 2009



DNI _____ Nombre _____ Apellidos _____
Titulación: ☐ Gestión ☐ Sistemas

3. (2 puntos) Se pretende resolver mediante la técnica de backtracking el problema de los eslabones. En este problema, se pretende construir una cadena de longitud n cm, a partir de eslabones de 4 longitudes diferentes (disponibles en número ilimitado), que se encuentran almacenados en un vector de enteros de 4 posiciones. Se pretende que el número de eslabones utilizado sea el mínimo posible.

a) (1,5 puntos) Completar el código Java para dar solución a este problema (escribirlo en los recuadros preparados a tal efecto).

```
public class Eslabones {
    private static int eslabones[] = { 1, 12, 5, 25 }; // Longitudes eslabones
    private int aux[] = new int[eslabones.length]; // solución parcial

    // Solución óptima
    private long nEslabonesOpt = 100000; // num. eslabones de la solución
    private int solOptima[] = new int[eslabones.length]; // de cada tipo

    public void buscarEslabones( ) {
        for ( ) {
            if ( ) {
                aux[i]++;
                if ( ) {
                    buscarEslabones( );
                }
                else {
                    if ( ) {
                        nEslabonesOpt = cEslab;
                        solOptima = (int[]) aux.clone();
                    }
                }
                aux[i]--;
            }
        }
    }

    public static void main(String[] args) {
        Eslabones es= new Eslabones();
        es.buscarEslabones(...); // aquí irían los parámetros correspondientes
    }
}
```

b) (0,5 puntos) Queremos optimizar el árbol de estados desarrollados realizando podas cuando sea posible. Escribir, en lenguaje natural, alguna posibilidad de poda del árbol, especificando: si es necesario algún requisito previo, en base a qué se puede realizar y cuándo se realizaría.

4. (1 punto) Se quiere resolver de una forma más eficiente el problema de los eslabones propuesto en el ejercicio 3, para ello se plantea un algoritmo voraz.

- Diseñar y expresar en lenguaje natural un heurístico para este algoritmo voraz.
- Para el heurístico diseñado, razonar si es óptimo o no, y si no lo es demostrarlo.

5. (1,25 puntos) El problema de la mochila 0/1 consiste en que disponemos de n objetos y una “mochila” para transportarlos. Cada objeto $i = 1, 2, \dots, n$ tiene un peso w_i y un valor v_i . La mochila puede llevar un peso que no sobrepase W . Los objetos *no* se pueden fragmentar: o tomamos un objeto completo o lo dejamos. El objetivo del problema es maximizar valor de los objetos respetando la limitación de peso.

La función que nos proporciona la solución al problema es la siguiente:

$$V(i, j) = \begin{cases} -\infty & \text{si } j < 0 \\ 0 & \text{si } i = 0 \text{ y } j \geq 0 \\ \max(V(i-1, j), V(i-1, j-w_i) + v_i) & \text{en otro caso} \end{cases}$$

Donde i , representa a los objetos y j , representa el peso de la mochila.

Utilizaremos la técnica de Programación Dinámica para obtener la solución óptima al problema.

- Representar la tabla necesaria para almacenar los valores intermedios (sin rellenar). Sabiendo que tenemos tres objetos y el peso máximo que puede llevar la mochila es 5.
- Explicar que valores podemos rellenar de forma directa en la tabla.
- Representar gráficamente sobre la tabla del apartado a) el patrón de dependencia de una celda cualquiera, es decir, marcar las celdas que son necesarias para calcular una celda dada. Se pueden plantear las hipótesis que sean necesarias sobre el valor y peso de los objetos.
- (0,5 puntos) Resolver este problema, escribiendo todos los valores sobre la tabla, para los siguientes objetos considerando que el peso máximo de la mochila es 5.

Obj.	1	2	3
w_i	2	3	2
v_i	4	7	5

6. (1,5 puntos) Cuestiones sobre la técnica de ramificación y poda (responder brevemente a lo que se plantea en cada apartado).

- Se pretende resolver un problema de optimización con un algoritmo de ramificación y poda pero sólo disponemos de un heurístico de ramificación. Razonar en qué casos este algoritmo es claramente más eficiente que backtracking y en cuáles no.
- Qué modificaciones hay que realizar en el algoritmo de recorrido en anchura para convertirlo en un recorrido en profundidad.
- En el problema del puzle del cuaderno didáctico resuelto con ramificación y poda, se ven varias funciones que se aplican sobre los estados del problema. Indicar para cada una de ellos, si constituyen un heurístico o no, y si son un heurístico especificar de qué tipo:
 - Evaluación de estado alcanzable: El estado objetivo será alcanzable desde un estado si y sólo si $\text{Sumatorio}(\text{menor}(i)) + x$ es par. Donde x es 1 si la casilla vacía se encuentra en una de las casillas sombreadas del tablero y 0 si no es así.
 - Evaluación del número de fichas colocadas en la situación final del estado actual.
 - $\text{Sumatorio}(\text{distancia}(i))$ desde 1 a 16, donde $\text{distancia}(i)$ es el número de movimientos necesarios para colocar la ficha i en la posición i .
 - Comprobación de estado repetido.

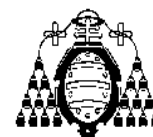


Examen de Teoría de la Programación

E. U. ING. TEC. EN INFORMÁTICA DE OVIEDO

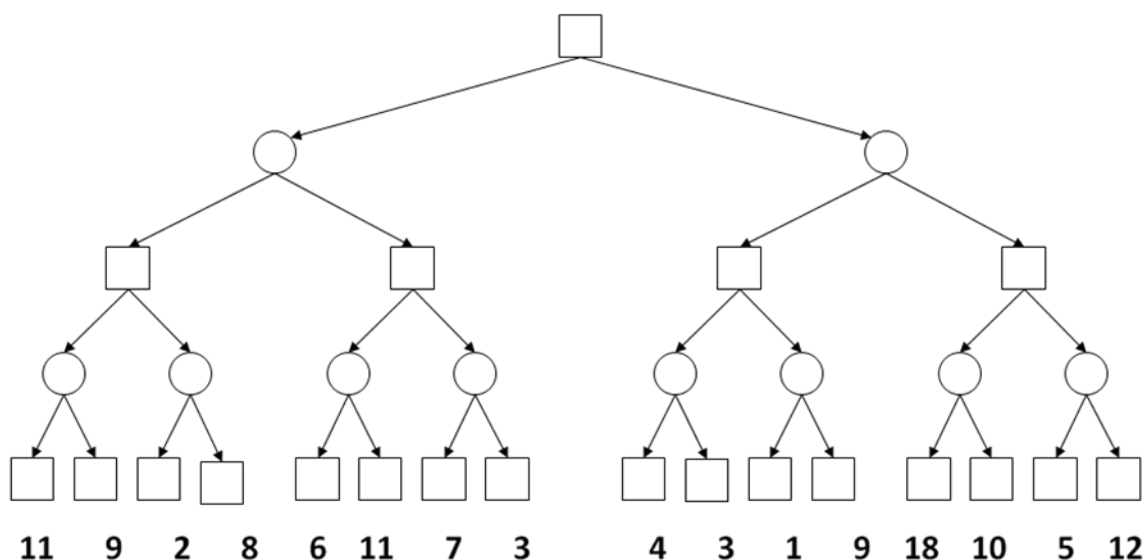
Final Febrero – Curso 2008-2009

7 de febrero de 2009



DNI _____ Nombre _____ Apellidos _____
Titulación: ☐ Gestión ☐ Sistemas

7. (1,25 puntos) Desarrollar la poda α - β para conocer que jugada debe realizar el jugador MAX, sobre el siguiente árbol:



- Sombrear los nodos que haya que desarrollar
- Escribir las cotas α y β ,
- Marcar los cortes e indicar si son de tipo α o β ,
- Por último, indicar que jugada debe elegir MAX para situarse en la mejor posición posible.

Notas: El jugador que realiza el primer movimiento en el árbol es MAX. Los nodos del árbol se desarrollan de izquierda a derecha.

Ejemplo de indicaciones:

