Examen de Teoría de la Programación

ESCUELA DE INGENIERÍA INFORMÁTICA – UNIVERSIDAD DE OVIEDO

Final Junio – Curso 2010-2011 28 de mayo de 2011

DNI _____ Nombre _____ Apellidos _____

Titulación: ☐ Gestión ☐ Sistemas

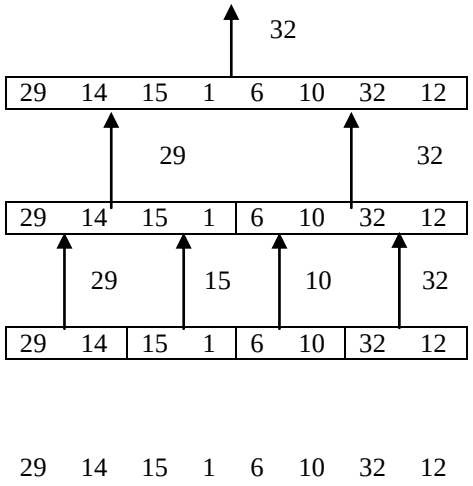
3. (2 puntos) Estamos desarrollando un módulo software para las cajas de un supermercado, en el que tenemos que dar información detallada sobre la cantidad de monedas de cada tipo (sistema monetario del euro) que hay que devolver cuando un cliente paga su compra.
Queremos que el número de monedas que manejan las cajeras sea siempre el mínimo posible, como el número total de monedas nunca será muy alto no nos importa demasiado el tiempo de cálculo. Hay que tener en cuenta que cada mañana, las cajeras meten un número fijo de 10 monedas para cada tipo y que en un momento determinado puede quedarse sin algún tipo de monedas.

a) Completar el código en Java del algoritmo que a partir de las monedas disponibles en ese momento y la cantidad que hay que devolver, permita obtener siempre el número óptimo de monedas o un error en caso de que no sea posible devolver la cantidad. Escribir el código de llamada al método.

```
public class Cambio {
       // constantes para el sistema monetario euro
       private static final float[] valor= {2, 1, 0.5F, 0.2F, 0.1F, 0.05F, 0.02F,
0.01F };

       private int[] cambio;
       private int numMonedas= 0;
       private int[] solucion;
       private int minMonedas= 10*8; // monedas iniciales

       […]

       public void devolverCambio(double importeDevolver, int[] disponible)
       {
         for (int i= 0; i<valor.length; i++)
           // Comprobar si estado es valido
           if ((disponible[i]>0) && (importeDevolver>=valor[i]))
           {
                // Anotar el estado
               importeDevolver= importeDevolver - valor[i];
                // decrementa lo que queda por devolver
                disponible[i]--;  // una moneda menos de ese valor
                cambio[i]++;            // mete moneda en la solución
                numMonedas++;

             if (importeDevolver>0.01F)
                    devolverCambio(importeDevolver, disponible);
             else
              // quedarse con la mejor solución hasta el momento
                    if (numMonedas<minMonedas)
                    {
                           solucion= cambio.clone();
                           minMonedas= numMonedas;
                    }

             // Borrar anotación de estado
               importeDevolver+= valor[i];
                disponible[i]++;
                cambio[i]--;
                numMonedas--;
           }

       }
}
```

4. (1,5 puntos) Tenemos un conjunto *S* de *n* números enteros. Queremos buscar el máximo de este conjunto utilizando la técnica de Divide y vencerás.

a) (1 punto) Dado el siguiente conjunto de 8 números. Representar gráficamente (utilizando los números que se representan abajo) las divisiones que se realizan en el vector en el esquema recursivo de división (rodeando con una línea los números que quedan en cada división) y el resultado que devolvería cada llamada recursiva (con una flecha hacia el vector anterior).



a) (0,5 puntos) ¿Qué complejidad tiene este algoritmo? Justifica el resultado.

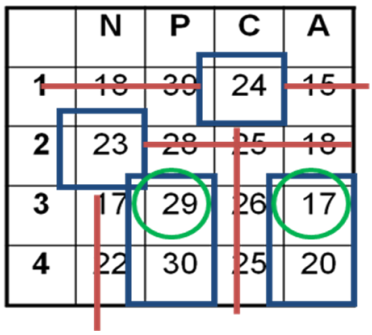
DV con división
a= 2
b= 2
K= 0
a= 2 > $b^K=2^0=1$ → $O(n^{\log_b a})=O(n)$

5. (2 puntos) El cuadriatlón es una variante de triatlón en la que se combinan Natación(N)-Piragua(P)-Ciclismo(C)-Atletismo(A). Ante la proximidad de los juegos olímpicos de Pekín 2008 (aunque ésta no es una prueba olímpica) se ha propuesto, a modo de entrenamiento para los deportistas, hacer una competición. Se ha decidido que la prueba sea por relevos y cada participante realizará un deporte y le dará el relevo a su compañero. El seleccionador dispone de cuatro deportistas y de sus tiempos para cada una de las pruebas. Se quiere diseñar el algoritmo que mediante la técnica de ramificación y poda permita decidir al seleccionador que deportista realizará cada prueba.
Deportista 1 (18-39-24-15). Deportista 2 (23-28-25-18). Deportista 3 (17-29-26-17). Deportista 4 (22-30-25-20). Los tiempos están en el mismo orden en el que se realizan los deportes.

a) (0,25 puntos) Explicar cómo se calcula el heurístico de ramificación. Y aplicarlo al estado en el que asignamos al deportista 1 a ciclismo y al deportista 2 a natación (y quedan dos deportistas por asignar).
b) (0,25 puntos) Explicar cómo se calcula la cota inicial de poda y razonar cuándo se produce el cambio de esta cota.
c) (1 punto) Representar el árbol de estados después de haber expandido dos estados del árbol.
d) (0,5 puntos) Representar de forma ordenada los estados que quedan en la cola de prioridad en la situación descrita en el punto c).

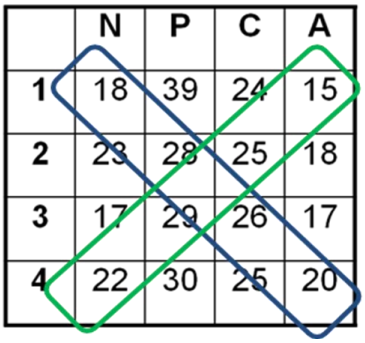
a) (0,25 puntos) Explicar cómo se calcula el heurístico de ramificación. Y aplicarlo al estado en el que asignamos al deportista 1 a ciclismo y al deportista 2 a natación (y quedan dos deportistas por asignar).

El cálculo del heurístico de ramificación consiste en: La suma de lo ya asignado en cada estado más el caso mejor (mínimo de columnas) de lo que queda por asignar.
Al estado 1→C, 2→N, le corresponde un coste de 24+23+29+17=93.

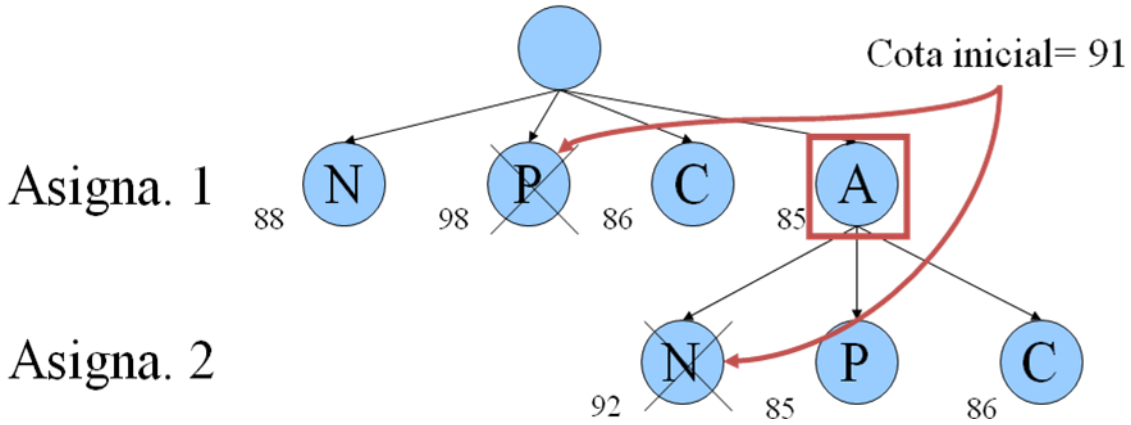


b) (0,25 puntos) Explicar cómo se calcula la cota inicial de poda y razonar cuándo se produce el cambio de esta cota.

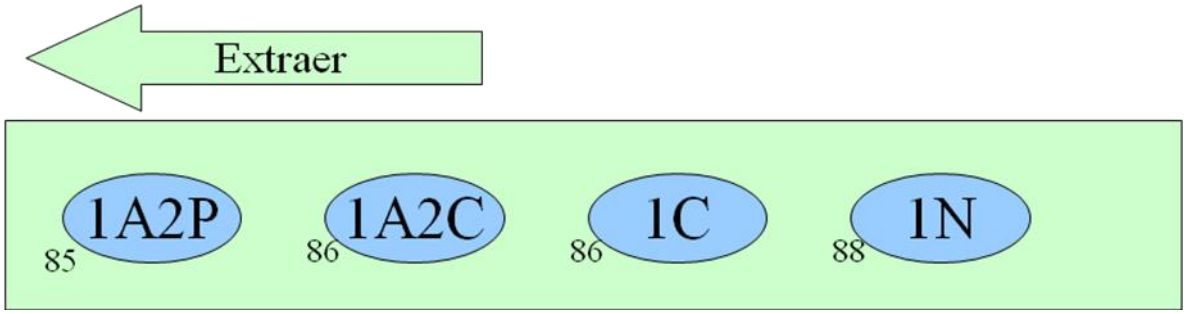
Inicialmente, es la menor de la sumas de las dos diagonales de la matriz de costes.
Valor cota inicial: min(91,92)=91.
Cuando en el desarrollo de los estados del árbol, lleguemos a una solución válida cuyo valor sea menor que el de la cota actual, se cambiará la cota a este nuevo valor.



c) (1 punto) Representar el árbol de estados después de haber expandido dos estados del árbol.



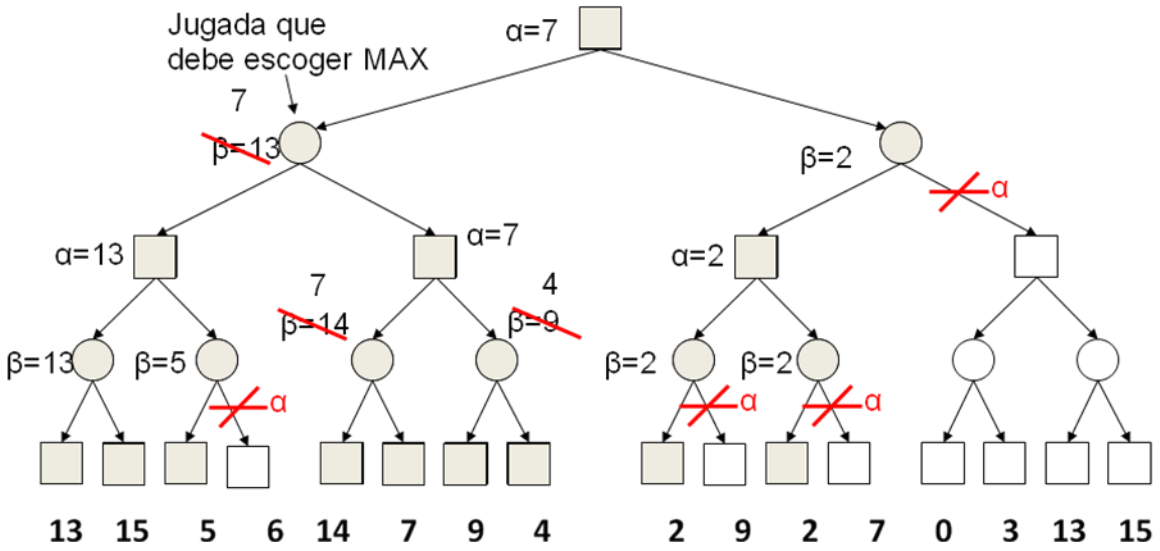
d) (0,5 puntos) Representar de forma ordenada los estados que quedan en la cola de prioridad en la situación descrita en el punto c).



DNI _____ Nombre _____ Apellidos _____

Titulación: ☐ Gestión ☐ Sistemas

6. (1,5 puntos) Desarrollar la poda α-β para conocer que jugada debe realizar el jugador MAX, sobre el siguiente árbol:



a) **Sombrear los nodos que haya que desarrollar**
b) **Escribir las cotas α y β,**
c) **Marcar los cortes e indicar si son de tipo α o β,**
d) **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.