



[Backtracking 7 puntos] La empresa ALG_APPS quiere realizar una aplicación para el juego del amigo invisible y nos encomienda su diseño. Los datos a manejar son varios. En primer lugar, el número de las personas que integran el juego, denominaremos P a ese número. En segundo lugar, se dispone de la información de los excluidos, esto es, se conoce si alguien no debe regalar a otra persona. Dicha información está recogida en el vector $\text{Excluir}[1..P]$, donde $\text{Excluir}[k]$ indica a qué persona no debe regalar la persona k . Los valores contenidos en dicho vector son valores comprendidos entre 0 y P . Indicar que si $\text{Excluir}[k]$ fuese 0, querría decir que la persona k puede regalar a cualquiera de los demás de la pandilla. Y por último, también se va a manejar el afecto que parece haber entre las personas que integran el juego. Dicha información se encuentra almacenada en la matriz $\text{Afecto}[1..P][1..P]$, donde $\text{Afecto}[i][j]$ es un número (≥ 0) que representa el afecto que tiene la persona i hacia la persona j , con $1 \leq i, j \leq P$.

Se pide diseñar un algoritmo de Vuelta_Atrás (*Backtracking*) que genere la forma de llevar a cabo el juego del amigo invisible, esto es, a quien debe regalar cada uno (teniendo en consideración todo lo relatado en el párrafo anterior) de manera que se maximice la suma de los afectos en los casos que haya un regalo de por medio, es decir, si resulta que la persona 3 tiene que regalar a la persona 7 entonces el afecto que se contabiliza es $\text{Afecto}[3][7]$.

Se pide responder con claridad y concisión a las siguientes cuestiones:

- Secuencia de decisiones (nº de decisiones y significado de las mismas) (10%)
- Función objetivo (10%)
- Restricciones explícitas (5%)
- Restricciones implícitas (15%)
- Tipo de solución buscada (Todas las factibles/Una factible/Óptima) (5%)
- Preparar_recorrido_nivel_k (5%)
- Existe_hermano_nivel_k (5%)
- Siguiente_hermano_nivel_k (5%)
- Función Solución (5%)
- Indica qué es lo que hacen las funciones Correcto y Valor si es que fueran necesarias en tu solución. Escribir el pseudocódigo de dichas funciones (\equiv cómo lo hacen) (20%)
- Dibujar el árbol de búsqueda que se explora hasta localizar la primera solución factible para el caso en el que $P=4$, esto es, hay 4 personas en el juego. Y donde $\text{Excluir}[1..4] = \{2, 0, 2, 0\}$. Numerar los nodos reflejando el orden en el que se visitan e indicar cuándo se realiza una poda (15%).

[Esquema voraz 3 puntos] Se pide resolver el mismo problema utilizando una estrategia voraz. Se deberá contestar de forma clara y concisa a las siguientes cuestiones: candidatos (10%), criterio de selección razonable y razonado (20%), función de factibilidad (10%), función de solución (10%) y mostrar, etapa a etapa, cómo la estrategia voraz descrita construye la solución para el siguiente ejemplo (50%):

$P = 4$, $\text{Excluir}[1..4] = \{2, 0, 2, 0\}$ y

	1	2	3	4
1	0	10	8	6
2	9	0	7	5
3	1	2	0	3
4	6	5	4	0



SOLUCIÓN.-

SECUENCIA DE DECISIONES

$\langle x_1, x_2, \dots, x_P \rangle$ donde x_i indica al amigo al que tiene que regalar la persona i .

FUNCIÓN OBJETIVO

$$\text{maximizar } \sum_{i=1}^P \text{Afecto}[i][x_i]$$

RESTRICCIONES EXPLÍCITAS

$$(\forall i)(x_i \in \{1, 2, \dots, P\}; 1 \leq i \leq P)$$

RESTRICCIONES IMPLÍCITAS

$$(\forall i) \left((\forall j)(i \neq j \rightarrow x_i \neq x_j : 1 \leq j \leq P) : 1 \leq i \leq P \right)$$

$$(\forall i)(i \neq x_i): 1 \leq i \leq P)$$

$$(\forall i)(x_i \neq \text{Excluir}[i]): 1 \leq i \leq P)$$

TIPO DE SOLUCIÓN BUSCADA

La solución óptima, por tanto esquema 3.

PREPARAR_RECORRIDO_NIVEL_K

$$x[k]=0$$

EXISTE_HERMANO_NIVEL_K

$$x[k] < P$$

SIGUIENTE_HERMANO_NIVEL_K

$$x[k] = x[k] + 1$$

FUNCIÓN SOLUCIÓN

$$k = P$$



FUNCIÓN CORRECTO

QUÉ HACE.- la función correcto, tras recibir la secuencia de decisiones x y el valor de k correspondiente, devuelve falso si la persona k se regala a sí misma o si la persona k regala a una persona a la que no debe regalar, esto es, a $\text{Excluir}[k]$. Además, si el amigo asignado a la persona k , esto es, $x[k]$, aparece en el tramo $[1..k-1]$ de la secuencia x de decisiones, la función devolverá falso. En cualquier otro caso, devolverá cierto.

PSEUDOCÓDIGO (CÓMO).-

Funcion Correcto ($\text{Excluir}[1..P]$:vector de enteros, x :tupla, k :entero) retorna (b :booleano)

```
var i:entero;ok:booleano fvar
i = 0;
ok = cierto;
si (  $k = x[k]$  )  $\vee$   $x[k] = \text{Excluir}[k]$  ) ok=falso; fsi
mientras (  $i < k-1 \wedge ok$  ) hacer
    i = i + 1;
    si (  $x[i] = x[k]$  ) entonces
        ok = falso;
    fsi
fmientras
retorna ok
ffunción
```

FUNCIÓN VALOR

QUÉ HACE.- la función valor, tras recibir la secuencia de decisiones x y el valor de k ($k=P$), devuelve el valor de la función objetivo correspondiente a la secuencia de decisiones x , esto es, la suma de los afectos conseguidos con la asignación de regalos.

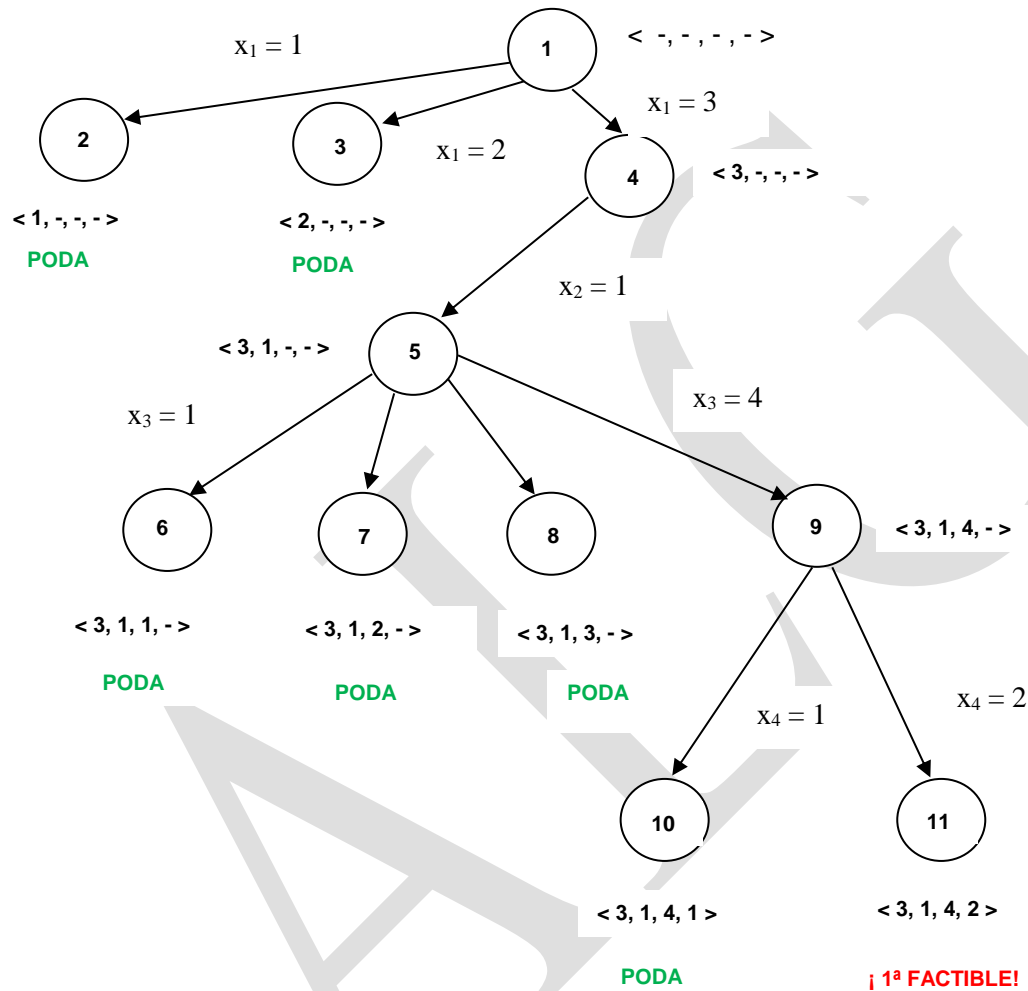
PSEUDOCÓDIGO (CÓMO).-

Funcion Valor ($\text{Afecto}[1..P][1..P]$: matriz de enteros, x :tupla, k :entero) retorna (b :entero)

```
var i, total: entero fvar
i = 0; total=0;
mientras (  $i \leq k-1$  ) hacer
    i=i+1
    total = total +  $\text{Afecto}[i][x[i]]$ 
fmientras
retorna total
ffunción
```



ÁRBOL DE BÚSQUEDA



- Nodo 2:** Poda porque la persona 1 no se puede regalar a sí misma
Nodo 3: Poda porque la persona 1 no puede regalar a la persona 2
Nodo 6: Poda porque la persona 1 no puede recibir más de un regalo
Nodo 7: Poda porque la persona 3 no puede regalar a la persona 2
Nodo 8: Poda porque la persona 3 no puede regalar a sí misma
Nodo 10: Poda porque la persona 1 no puede recibir más de un regalo



CANDIDATOS

Siguiendo la solución de backtracking, la secuencia de decisiones será $\langle x_1, x_2, \dots, x_P \rangle$ donde x_i indica el amigo al que regalará la persona i -ésima.

Los candidatos por tanto serán las P personas. Todas ellas formarán parte de la solución, la cuestión es en qué orden forman parte de la solución.

CRITERIO DE SELECCIÓN

En cada etapa i se seleccionará aquel amigo aún no seleccionado, distinto de la persona i -ésima y no excluido para la persona i ($\text{Excluir}[i]$) cuyo afecto de la persona i hacia él sea el máximo.

FUNCIÓN DE FACTIBILIDAD

El criterio de selección propuesto hace que todas decisiones sean factibles.

FUNCIÓN DE SOLUCIÓN

Habremos encontrado la solución cuando en el conjunto solución estén las P personas.

MOSTRAR ETAPA A ETAPA COMO LA ESTRATEGIA VORAZ CONSTRUYE LA SOLUCIÓN

$P = 4$, $\text{Excluir}[1..4] = \{ 2, 0, 2, 0 \}$ y

	1	2	3	4
1	0	10	8	6
2	9	0	7	5
3	1	2	0	3
4	6	5	4	0

Etapa	Candidato seleccionado	Solución	Valor total
inicial	--	$\langle -, -, -, - \rangle$	0
1	Amigo 3	$\langle 3, -, -, - \rangle$	$0+8=8$
2	Amigo 1	$\langle 3, 1, -, - \rangle$	$8+9=17$
3	Amigo 4	$\langle 3, 1, 4, - \rangle$	$17+3=20$
4	Amigo 2	$\langle 3, 1, 4, 2 \rangle$	$20+5=25$