



# PREGUNTA 3 – PED 1.

PREDAS. – UNED – LAS TABLAS. PAULINO ESTEBAN BERMÚDEZ  
RODRÍGUEZ

### 3.- CUESTIONES TEÓRICAS DE LA PRÁCTICA

1) ¿Cuál sería una manera no recursiva de realizar el problema? No es necesario exponer un algoritmo.

Realizándolo de manera iterativa usando dos bucles for() para rellenar las casillas.

[ ] vs [ ] Uno para la zona derecha y otro para la zona izquierda, usando dos contadores para llevar las vueltas realizadas por los bucles. El resultado obtenido sería como el siguiente.

```
/*
 * Para rellenar la tabla de competición usando el sistema de liga, necesito dibujarla de la siguiente manera:
 * Ej: N:7 --> Para saber quien descansa ese día a N le sumo 1. N+1 = 8
 * * GrupoX: --> N/2 -> 7/2 = 4 (Redondeo a la alza)
 * |
 * | N-1 -> 7-1 = 6
 * v
 * 1. |8_7|1_6|2_5|3_4| ==> 7 descansa 1 día
 * 2. |4_8|5_3|6_2|7_1| ==> 4 descansa 2 día
 * 3. |8_1|2_7|3_6|4_5| ==> 1 descansa 3 día
 * 4. |5_8|6_4|7_3|1_2| ==> 5 descansa 4 día
 * 5. |8_2|3_1|4_7|5_6| ==> 2 descansa 5 día
 * 6. |6_8|7_5|1_4|2_3| ==> 6 descansa 6 día
 * 7. |7_3|4_2|5_1|6_7| ==> 3 descansa 7 día
 * +-----+
 *
 * Ej2: N = 8 --> Al ser par, el torneo termina en n-1 días
 * * GrupoX: --> N/2 -> 8/2 = 4
 * |
 * | N-1 -> 8-1 = 7
 * v
 * 1. |7_8|1_6|2_5|3_4|
 * 2. |8_3|4_2|5_1|6_7|
 * 3. |6_8|7_5|1_4|2_3|
 * 4. |8_2|3_1|4_7|5_6|
 * 5. |5_8|6_4|7_3|1_2|
 * 6. |8_1|2_7|3_6|4_5|
 * 7. |4_8|5_3|6_2|7_1|
 * +-----+
 */
```

2) Expón la diferencia de coste (si la hubiera) entre ambas modalidades.

En el caso del algoritmo recursivo, el tamaño del problema es calcular la función de n, o que es lo mismo, el número de jugadores del torneo, para el caso del problema enunciado, que es:

Caso base

$$T(n) = K(n)$$

Para el caso de n, tenemos que el coste es:

$$T(n) \in \begin{cases} K(n) & \text{si } a = 2 \\ a * T(n') + K(n) & \text{si } a > 2 \end{cases} = \begin{cases} O(n * c_{nr}(n) + c_b(n)) & \text{si } a = 2 \\ O(a^{(n \div b)} * (c_{nr}(n) + c_b(n))) & \text{si } a > 2 \end{cases}$$

Los valores de a = 2 y b = n, x=n.

El coste en el caso recursivo es de  $O(2K)^2$ , pues la complejidad espacial está en la tabla, al rellenarla.

Para el caso iterativo, la definición del tamaño del problema es 'n', el tamaño del vector de entrada,

Las operaciones básicas (variables, asignaciones, valores de un array) son constantes,  $O(n)=1$ .

En las sentencias condicionales (if-else) tenemos un coste de  $O(n)=1$  al tener operaciones básicas que le acompañan, en este caso asignación de valores.

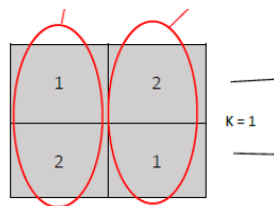
Para los bucles for, el producto de órdenes es el orden del producto  $O(f) * O(g) \rightarrow O(n) * O(n) = O(n^2)$ , al ser un doble bucle for, pues su coste es  $O(2*n)^2$ .

El coste en el caso iterativo es de  $O(2K)^2$ , pues la complejidad aumenta al tener que controlar el doble bucle.

### 3) Expón las opciones posibles para el caso trivial del algoritmo.

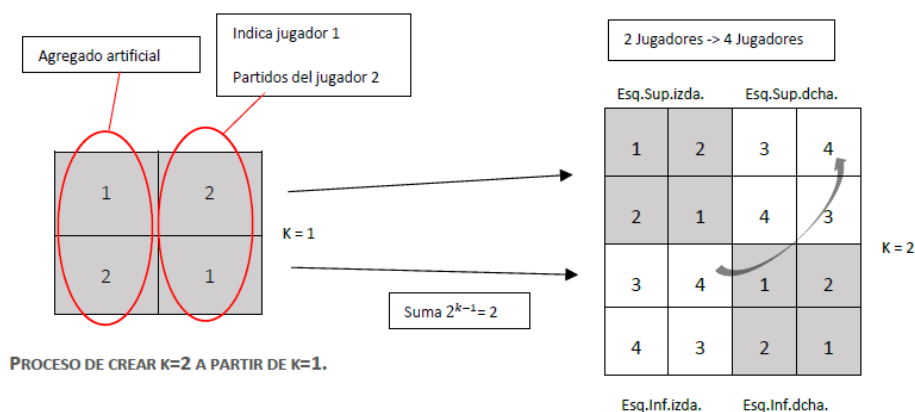
2

El caso base del problema, es la competencia del mínimo de jugadores de tenis posible, en este caso 2. Con estos dos jugadores, en tan solo 1 día ya puede enfrentarse ‘todos contra todos’, (así de sencillo).

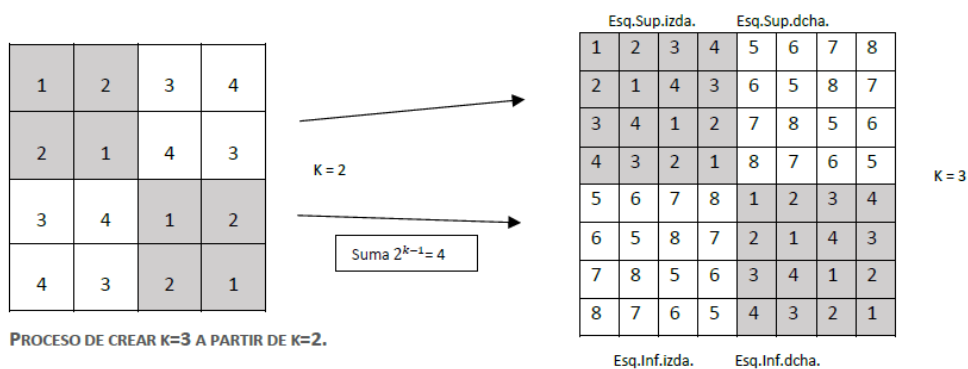


PROCESO DE CREAR K=2 A PARTIR DE K=1.

El siguiente caso, para 3/4 jugadores, debe aplicarse la misma técnica, si el jugador n juega contra n+1, para la celda de n+1, su adversario será n-1. Como la primera columna siempre representa los jugadores, llevamos los mismos datos obtenidos en los casos de las esquinas superiores a las esquinas inferiores, siguiendo el orden



PROCESO DE CREAR K=2 A PARTIR DE K=1.



PROCESO DE CREAR K=3 A PARTIR DE K=2.