

- Identifique qué parámetros debe tomar la función recursiva que resuelve el problema.
 - Describa con palabras **qué calcula** la misma, en función de sus argumentos.
 - Defina la función recursiva en notación matemática y opcionalmente en código.
 - Indique cuál es la llamada principal que obtiene el resultado pedido en el ejercicio.
9. El juego $\nwarrow \cup \uparrow \nearrow$ consiste en mover una ficha en un tablero de n filas por n columnas desde la fila inferior a la superior. La ficha se ubica al azar en una de las casillas de la fila inferior y en cada movimiento se desplaza a casillas adyacentes que estén en la fila superior a la actual, es decir, la ficha puede moverse a:
- la casilla que está inmediatamente arriba,
 - la casilla que está arriba y a la izquierda (si la ficha no está en la columna extrema izquierda),
 - la casilla que está arriba y a la derecha (si la ficha no está en la columna extrema derecha).

Cada casilla tiene asociado un número entero c_{ij} ($i, j = 1, \dots, n$) que indica el puntaje a asignar cuando la ficha esté en la casilla. El puntaje final se obtiene sumando el puntaje de todas las casillas recorridas por la ficha, incluyendo las de las filas superior e inferior.

Determinar el máximo y el mínimo puntaje que se puede obtener en el juego.

• Parámetros de la función recursiva

La función recursiva tomará **dos parámetros** para resolver el problema:

- f , que indica la fila del tablero en la que me encuentre.
- c , que indica la columna del tablero en la que me encuentre.

• ¿Qué calcula la función recursiva?

El ejercicio pide obtener dos funciones, una que calcule el máximo puntaje del juego y otra el mínimo. Entonces:

- La función **maxUP** calcula el puntaje máximo obtenible en el juego $\nwarrow \cup \uparrow \nearrow$ a partir de un tablero $n \times n$, donde n son la cantidad de filas y columnas, f es la fila donde se para el jugador para ver a la siguiente de arriba y c es una columna.
- La función **minUP** calcula el puntaje mínimo obtenible en el juego $\nwarrow \cup \uparrow \nearrow$ a partir de un tablero $n \times n$, donde n son la cantidad de filas y columnas, f es la fila donde se para el jugador para ver a la siguiente de arriba y c es una columna.

• Función en notación matemática

```
maxUP(f,c) = ( si f = 1                → c1,c
               | si 1 < f ≤ n ∧ c = 1  → cf,1 + maxUP(f-1,c) `max` cf,1 + maxUP(f-1,c+1)
               | si 1 < f ≤ n ∧ c = n  → cf,n + maxUP(f-1,c-1) `max` cf,n + maxUP(f-1,c)
               | si 1 < f ≤ n ∧ 1 < c < n → cf,c + maxUP(f-1,c-1) `max` cf,c + maxUP(f-1,c)
               | si 1 < f ≤ n ∧ 1 < c < n → `max` cf,c + maxUP(f-1,c+1)
               )
```

Donde:

- Si está en la primera fila significa que no puedo subir más (fin del juego) y devuelve simplemente el valor de la casilla.
- Si se está parado en la primera columna, elijo el máximo puntaje entre subir a la fila de arriba en la misma columna (\uparrow) y la columna siguiente (\nearrow).
- Si se está parado en la última columna, elijo el máximo puntaje entre subir a la fila de arriba en la misma columna (\uparrow) y la columna anterior (\nwarrow).
- Si no estoy en alguna columna borde, entonces elijo el máximo puntaje entre subir a la fila de arriba en la columna anterior (\nwarrow), en la misma columna (\uparrow) y la columna siguiente (\nearrow).

```

minUP(f,c) = ( si f = 1                → c1,c
               | si 1 < f ≤ n ∧ c = 1  → cf,1 + minUP(f-1,c) `min` cf,1 + minUP(f-1,c+1)
               | si 1 < f ≤ n ∧ c = n  → cf,n + minUP(f-1,c-1) `min` cf,n + minUP(f-1,c)
               | si 1 < f ≤ n ∧ 1 < c < n → cf,c + minUP(f-1,c-1) `min` cf,c + minUP(f-1,c)
               | si 1 < f ≤ n ∧ 1 < c < n → `min` cf,c + minUP(f-1,c+1)
               )

```

Donde:

- Si está en la primera fila significa que no puedo subir más (fin del juego) y devuelve simplemente el valor de la casilla.
- Si se está parado en la primera columna, elijo el mínimo puntaje entre subir a la fila de arriba en la misma columna y la columna siguiente.
- Si se está parado en la última columna, elijo el mínimo puntaje entre subir a la fila de arriba en la misma columna y la columna anterior.
- Si no estoy en alguna columna borde, entonces elijo el mínimo puntaje entre subir a la fila de arriba en la columna anterior, en la misma columna y la columna siguiente.

• Llamada principal

La llamada principal es ***maxUP(n,j)*** y ***minUP(n,j)***, pues el juego empieza en la última fila n y va subiendo hasta la primera. No obstante, necesito ver todas las posibilidades partiendo desde cada una de las columnas. La solución a esto es utilizar un bucle e ir comparando los puntajes de cada columna.

```

var max_puntaje: nat
max_puntaje := -∞

for j := 1 to n do
  max_puntaje := max_puntaje `max` maxUP(n,j)
od

```

```

var min_puntaje: nat
min_puntaje := ∞

for j := 1 to n do
  min_puntaje := min_puntaje `min` minUP(n,j)
od

```