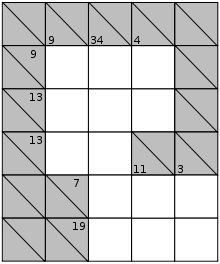
Kakuro

# Explicación del proyecto

El Kakuro se es un rompecabezas que consiste en una cuadrícula, algunos de los cuales tendremos que rellenar con números. Una de las reglas es que esos números sólo pueden ser del 1 al 9.

Aquí tenemos un ejemplo muy simple. En este caso los espacios en blanco corresponden a los espacios donde hay que introducir los números. Estos no pueden repetirse en ni por filas ni por columnas.

Como podemos ver, también hay unos números en el ejemplo. Esta es la verdadera dificultad del rompecabezas, ya que indican el valor que deben sumar los números que pertenecen a una determinada columna o fila.

Por ejemplo, en la columna inferior derecha hay una columna con valor tres. Teniendo en cuenta que hay dos casillas, que no se pueden repetir valores, y que los valores deben ser entre el 1 y el 9, las únicas posibilidades es que esa columna tenga valores (1,2) o (2,1).



En primer lugar, cuando llamamos a nuestra función ‘kakuro’ necesitamos tres valores. El primero (‘VALS’) corresponde a nuestra matriz de números, en los que los espacios en los que no se introducen números se simbolizan con un 0. Las otras dos variables corresponden a la suma en horizontal (‘SUMX’) y en vertical (‘SUMY’). En el caso que se llame a esta función se calcularán que los valores de la matriz cumplen con las sumas horizontales y posteriormente se comprobará de igual manera las sumas verticales haciendo la traspuesta de la matriz.

La comprobación de las filas requiere cuatro valores. En primer lugar, la matriz a comprobar y el segundo corresponde a los valores de las sumas. Los dos siguientes son acumuladores, en el que el primero de ellos corresponde a la suma que se va realizando con los valores que se encuentran de una fila (que se pondrá a cero cuando acabe) y, en segundo lugar, un array de los números disponibles, para evitar que se repitan en una misma fila.

Para comprobar cada fila comprobaremos siempre el primer valor de la primera fila. Hay varias opciones a contemplar. En primer lugar, podría haber un número del 1 al 9, o una variable. En nuestro caso obligamos a que si hay una variable tenga que haber un nº del 1 al 9. La segunda opción podría haber un 0, lo que indicaría que hay un espacio en el que no se introduce ningún número y que posiblemente sea un final de ‘subfila’. Por último, se contempla la posibilidad de que ya esa fila esté vacía, en tal caso también podría ser un final de ‘subfila’ o no.

En la primera opción, utilizamos la función selecciona para comprobar si el primer elemento pertenece a uno de los posibles y en tal caso obtener un nuevo array que elimine dicho elemento. Al haber un número, el acumulador debe aumentar, por lo que se le suma el valor del número asignado a dicha casilla. Por último, se vuelve a llamar a la función con el nuevo acumulador, los nuevos números disponibles y quitando el primer valor de la primera fila ya que lo acabamos de comprobar.

La segunda se subdivide en otras dos opciones. Una de ellas sería que el acumulador valiera 0, lo que significaría que no ha habido números antes, por lo que simplemente quitamos el primer valor de la matriz. Primero comprobamos que el primer valor es una constante, y no una variable. Con esto lo que queremos conseguir es que, si se utiliza una variable para conocer el valor que podría ser, no contemple la posibilidad de que sea una casilla no apta.

Ahora, lo que tenemos que tener en cuenta es si cuando encontramos un 0 quiere decir que ha terminado una fila. Para ello, comprobamos que el primer valor de la suma corresponde al acumulador y que este valor no sea cero. Si cumple estas condiciones se volverá a llamar a la función quitando el primer valor de la matriz, el primer valor de la suma porque ya lo hemos comprobando y el acumulador y los valores disponibles se resetean porque se ha acabado una fila.

Finalmente, una comprobación acaba correctamente si la matriz y el array de las sumas se vacían al mismo tiempo.

# Proceso de realización

Entre las dificultades, el mayor contratiempo fue darnos cuentas de que utilizar variables podría dificultar orientar el árbol por una rama concreta. Para simular los espacios en los que no se introducen números utilizamos el valor 0. Cuando queremos detectar que

# División de roles

En primer lugar, ambos miembros del grupo plantearon una visión inicial del problema y aportar a la realización del informe.

Posteriormente la división se realizó de la siguiente forma:

Jorge González Cabrera

* Resolución del rompecabezas.

Sonia Díaz Santos

* Realización de la traspuesta de una matriz.
* Mostrar el resultado formateado por pantalla.

Dudas:

-Si es suficiente

-Si el rendimiento es suficiente

-Formato de informe

-Método write