



# Proyecto Final

## FUNDAMENTOS DE ANÁLISIS Y DISEÑO DE ALGORITMOS

Carlos Andres Delgado S, Ing \*

Mayo de 2017

Este taller se puede trabajar en grupos de hasta 3 personas. Entregue el código fuente solicitado y elabore un informe en formato PDF con la información solicitada en este proyecto.

### 1. El problema

Como si no fueran lo suficientemente complejos los Sudokus, en Julio de 2009, la revista Games Magazine describió una variante del juego que combina el Sudoku y el Dominó llamado **Su-Domino-Ku**. El juego tiene la forma de un Sudoku normal, es decir, una grilla de 99 que debe ser llenada usando solo los dígitos del 1 al 9. En una solución exitosa:

- Cada fila tiene que contener todos los del 1 al 9.
- Cada columna tiene que contener todos los dígitos del 1 al 9.
- Cada región de 3 x 3 tiene que contener todos los dígitos del 1 al 9.

Pero para un Su-Domino-ku, nueve celdas son inicializadas con los números del 1 al 9. Esto deja 72 celdas sin inicializar. Ellas deben ser llenadas usando un conjunto de 36 piezas de dominó. El conjunto incluye un dominó para cada posible par de números únicos desde el 1 hasta el 9 (Ejemplo: 1+2, 1+3, 1+4, 1+5, 1+6, 1+7, 1+8, 1+9, 2+3, 2+4, 2+5, ...). Note que no existen las piezas simétricas 1+2 y 2+1 en el conjunto, ya que cada una de estas piezas puede convertirse en la otra rotándola. Además note que los dominós pueden cruzar el límite de las regiones 3x3 (como lo hace el dominó 9 + 2 del siguiente ejemplo). Por ejemplo, la siguiente figura 1 muestra un juego en su estado inicial y la figura 2 la única forma de completar el rompecabezas:

	1	2	3	4	5	6	7	8	9
A		7	2						5
B									
C					1	8			
D									3
E					6				
F									
G									
H									
I	9			4					

Figura 1: Su-Domino-ku en su estado inicial

	1	2	3	4	5	6	7	8	9
A	8	7	2	6	4	3	1	9	5
B	3	6	1	9	7	5	8	4	2
C	5	4	9	2	1	8	6	3	7
D	1	2	6	7	5	4	9	8	3
E	7	3	8	1	6	9	2	5	4
F	4	9	5	8	3	2	7	6	1
G	2	8	4	5	9	7	3	1	6
H	6	5	7	3	8	1	4	2	9
I	9	1	3	4	2	6	5	7	8

Figura 2: Su-Domino-ku en su estado final

Se espera que a partir de una entrada dada, el programa calcule la solución del Su-Domino-Ku. Las fichas disponibles se muestran en la siguiente figura:

\*carlos.andres.delgado@correounivalle.edu.co

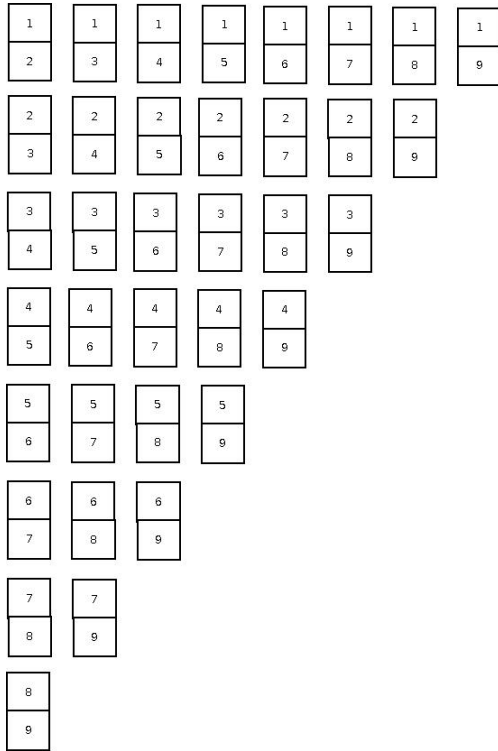


Figura 3: Fichas de dominó

Recuerde la rotación de una ficha:

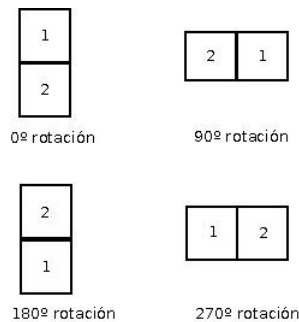


Figura 4: Rotaciones de una ficha de dominó

## 2. Solución

Para el desarrollo de este proyecto se pueden escoger el ambiente de desarrollo y el lenguaje de programación en el que va a trabajar.

El estado inicial del tablero se ingresa en un archivo de siguiente forma:

- 9 filas, donde cada final tiene el siguiente formato:
  - Primera columna indica el valor a insertar.
  - El número de la fila donde va el valor.
  - El número de la columna donde va el valor.

- La décima fila contiene el número de fichas ya insertadas en el tablero. Este valor es mayor o igual 0. En el caso que es 0 no hay fichas en el tablero.
- Las siguientes filas, indican en sus dos primeras columnas la ficha de la figura 1, en la tercera su rotación y en las dos siguientes su posición inicial en el tablero.

Para el estado inicial mostrado en la figura la entrada tendría la siguiente forma:

```

1 3 5
2 1 3
3 4 9
4 9 4
5 1 9
6 5 5
7 1 2
8 3 6
9 9 1
10
1 6 90 2 2
4 8 90 2 8
2 7 0 2 9
2 9 90 3 4
6 8 0 3 3
4 7 180 6 1
2 3 90 6 6
6 7 90 6 8
5 9 270 7 5
7 8 270 9 9

```

La salida se debe mostrar o almacenar en un archivo de salida de la siguiente forma:

- 9 filas, con cada una 9 columnas que muestra la solución final del Su-Domino-Ku.
- La décima fila es un espacio en blanco.
- De la 11 fila en adelante con las figuras utilizadas, indicando su valor, rotación y posición. En caso de que la ficha no esté siendo utilizada en la solución, no se coloca rotación ni posición de la ficha.

Un fragmento de la salida del algoritmo se puede ver a continuación.

```

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

1 2 0 7 8
1 3 270 9 3
1 4 180 5 9
1 5 180 1 3
1 6 180 2 2
1 7 180 7 6
1 8 0 5 4
1 9 90 1 8
...

```

### 3. Criterios de evaluación

Se deben implementar dos soluciones utilizando técnicas de programación diferentes: Programación Dinámica y Programación Voraz.

- Se debe documentar cada una de las soluciones escogidas, explicando la lógica de solución.
- Justificación de las estructuras de datos utilizadas en su solución.
- Se debe estimar la eficiencia en tiempo de ejecución de los dos algoritmos implementados.
- Se deben presentar las conclusiones del trabajo desarrollado.
- Este proyecto debe ser sustentado. La nota de sustentación va entre 0 y 1 es individual y su nota final se calcula multiplicando este factor por la nota obtenida en el proyecto.

Los criterios de evaluación son:

1. **(30 %)** Planteamiento completo de la solución dinámica
  - a) Caracterización de la estructura optima
  - b) Definición recursiva del valor de una solución óptima
  - c) Calculo de una solución óptima de manera Bottom-up
  - d) Construcción de la solución óptima a partir de la información calculada
  - e) Análisis de complejidad
2. **(30 %)** Planteamiento de la solución voraz y su análisis de complejidad.
3. **(15 %)** Redacción y estructura del informe. Incluya conclusiones, las cuales no pueden ser cosas del marco teórico
4. **(25 %)** Implementación. Es de aclarar que debe cumplir:
  - Carga de entradas. El usuario puede especificar el archivo que puede cargar
  - Implementación de estrategia voraz
  - Implementación de estrategia de programación dinámica
  - Desplegar al usuario información sobre la solución al problema (tablero)
  - Desplegar al usuario información sobre la solución al problema (ingreso de fichas)