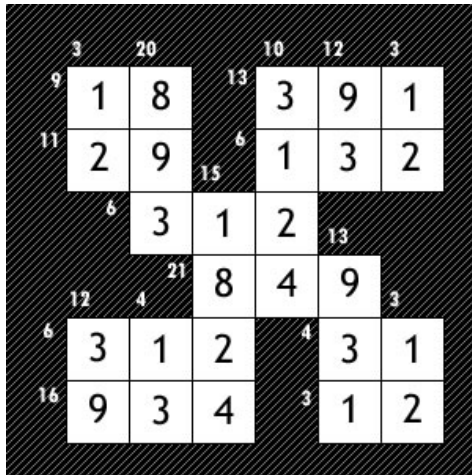


Proyecto 2 - Kakuro (カックロ)



Un kakuro es un tipo de sudoku-crucigrama. Consta de casillas con numeros clave que indican cuanto deben sumar los numeros a llenar hacia abajo y hacia la derecha. Otras casillas en blanco deben ser llenadas con numeros que sumen el numero que se indica. Las casillas negras tienen valores de 0.

Los únicos números que se pueden usar para llenar los espacios son del 1 al 9, sin repetir ningún numero en la fila o en la columna.

Backtracking

Se debe implementar una solución que genere kakuros de tamaño 14x14. Así mismo se debe programar un algoritmo **basado en backtracking** para solucionar un kakuro dado. El programa debe poder salvar los kakuros generados en un archivo y cargarlos de vuelta para ser resueltos automáticamente.

Poda

Se debe crear una o varias funciones que calculen si una solución parcial es prometedora o no. Esta función es clave en el programa para evitar que se exploren todas las posibles y por ende, hacer que el programa termine de generar o solucionar un kakuro en tiempo adecuado.

Interfaz

Se debe crear una interfaz de usuario que muestre la solución de un kakuro y también el kakuro generado.

Hilos y Forks

Se debe poder escoger si buscar la solución de un kakuro o generarlo usando hilos o forks, o ninguno. La idea es mejorar los tiempos de respuesta de la aplicación usando paralelización. Se debe de poder poner un tope a la cantidad de forks e hilos, y que si se llega a ese tope, no se generen mas que esos, siempre manteniendo el tope. En el momento que hay menos hilos del tope, se pueden generar mas, siempre respetando el tope.

Lenguaje de Programación

Instituto Tecnológico de Costa Rica
Escuela de Ingeniería en Computación
Curso: Análisis de Algoritmos
Profesor: José Carranza-Rojas
Valor: 15%
Proyecto en parejas

Semestre II, 2017

Se debe programar en Python.

Documento en Latex / PDF

Deben crear en latex (y entregar los fuentes de latex) un documento donde describen el análisis del algoritmo de backtracking para generar y para solucionar kakuros, así como el análisis del algoritmo que permuta números, y el algoritmo usado para podar el árbol implícito. El documento debe seguir el template de la IEEE para publicaciones en ingeniería el cual pueden encontrar aquí:

<https://www.ieee.org/documents/IEEEtran.zip>

El documento además tendrá un análisis comparativo (experimentos) para entender la duración en milisegundos de correr el algoritmo de solución de kakuros sin paralelización, versus la versión con hilos y con forks. Se deben definir corridas con forks y con hilos y observar que posiblemente a partir de cierto número de hilos/forks la eficiencia se degrada.



Evaluación

Tarea	Puntaje Máximo
Programacion Backtracking	15
Programacion Forks	10
Programacion Hilos	10
Solucion de poda (inventiva, programacion)	15
Analisis O grande de funcion de poda	10
Analisis O grande permutaciones	5
Analisis O grande backtracking	10
Experimentos y discusión	15
Interfaz	5
Manejo de archivos	5