



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

Practica 01: Problema de adoquinamiento

ALUMNO

Kevin Antonio Carvajal - 319160242

PROFESOR

María de Luz Gasca Soto

ASIGNATURA

Análisis de Algoritmos I 7156

12 de septiembre de 2023

1. Introducción:

El problema de adoquinamiento consiste en cubrir completamente una superficie cuadrada de tamaño $n \times n$ con adoquines en forma de L. La cuadrícula inicial comienza con un adoquín especial que ocupa una sola localidad en la matriz y no puede ser tapado por algún adoquín en forma de L. Este informe detalla cómo implementamos una solución para este problema utilizando el lenguaje de programación Python.

Algoritmo solución:

- a) Se crea una matriz de tamaño $n \times n$ inicializada con ceros, que representa la cuadrícula. En la matriz, cada celda representa una ubicación en la cuadrícula que queremos llenar con adoquines.
- b) Se optó por colocar el adoquín especial en la esquina superior izquierda de la matriz. El valor -1 se utilizó para marcar esta ubicación especial en la matriz. La elección del valor -1 como marcador es porque permite identificarlo fácilmente y distinguirlo de los adoquines en forma de L que se colocarán posteriormente, que generalmente se representan con números enteros positivos.
- c) Para llevar un registro de cuántos adoquines hemos colocado en la cuadrícula, utilizamos un contador global, que aumenta cada vez que colocamos un adoquín. Cada adoquín se representa en la matriz como un número entero positivo. De esta manera, cada adoquín tiene un número único.
- d) La función `fillRegion` se utiliza para llenar una región específica de la cuadrícula con adoquines en forma de L. Si la región es lo suficientemente pequeña (caso base 2×2), se llena directamente con adoquines en forma de L. Si la región es más grande, se divide en cuatro partes más pequeñas y se llama recursivamente a `fillRegion` para llenar cada una de esas partes. La recursión se aplica hasta que las regiones sean lo suficientemente pequeñas para llenarlas directamente.

Ejecutar el Programa:

- a) Abra una terminal.
- b) Navegue a la carpeta donde se encuentra el archivo `main.py` que contiene el código fuente.
- c) Ejecute el programa proporcionando un valor entero positivo k como argumento. Este valor k determinará el tamaño de la cuadrícula $n \times n$, donde $n = 2^k$.
- d) Ejemplo de ejecución: `python main.py 3`

La implementación exitosa de este programa en Python resuelve el problema de adoquinamiento utilizando adoquines en forma de L. El programa toma como entrada el tamaño de la cuadrícula y genera una representación visual de la solución en la terminal.