

Universidad Nacional Autónoma de México

FACULTAD DE CIENCIAS

ANÁLISIS DE ALGORITMOS

Práctica 01
Problema de Adoquinamiento

María de Luz Gasca Soto
Teresa Becerril Torres
Rodrigo Fernando Velázquez Cruz

Autor:

Hermes Alberto Delgado Díaz



03 de septiembre del 2024

La solución al problema de adoquinamiento se representa mediante una matriz bidimensional llamada '**tablero**' y un algoritmo recursivo basado en la técnica divide y vencéras.

Esta técnica dice que un problema lo podemos dividir en subproblemas hasta que llegue a ser lo suficientemente sencillo de solucionar.

En el caso del problema de adoquinamiento el tablero se divide en 4 subtableros recursivamente, hasta llegar al subtablero mas sencillo para resolver.

El método **colocarAdoquines** es el responsable de dividir el tablero en subtableros más pequeños y de colocar los adoquines L en el tablero.

Para colocar los adoquines L se hace llamado al metodo **colocar** el cual mediante coordenadas coloca un adquin L en el tablero mas grande.

El proceso de división del tablero es el siguiente

1. Para valores de $n > 2$, el tablero se divide en 4 subtableros de tamaño $n/2 \times n/2$.
2. La celda ocupada en el subtablero actual se localiza y, dependiendo de su posición(primer, segundo, tercer, cuarto subtablero), se coloca un adquin L para cubrir las celdas centrales de los otros tres cuadrantes.
3. El metodo **colocarAdoquines** se llama para cada unos de los subtableros y se repite el proceso.

Lenguaje de programación utilizado

Para esta práctica se utilizo java en su versión **java 22.0.2**

Comandos

Para compilar el programa

```
?- javac Adoquin.java
```

Para ejecutar el programa

```
?- java Adoquin <k>
```

k es el tamaño que se desea del tablero en formato $2^k \times 2^k$, k debe ser entero.

Ejemplos de ejecución

Cuando $k = 1$

```
?- javac Adoquin.java
?- java Adoquin 1
Adoquin especial en la posicion (0,0) con valor -1
-1  1
 1  1
```

El adoquín especial siempre se representa con valor -1 , cada adoquín L que se coloca se le asigna un número distinto para poder diferenciarlos.

Cuando $k = 2$

```
?- javac Adoquin.java
?- java Adoquin 2
Adoquin especial en la posicion (0,0) con valor -1
-1  3  2  2
 3  3  1  2
 4  1  1  5
 4  4  5  5
```