

**Tarea 1 - Análisis de Algoritmos y Estructura de Datos**  
**Leonardo Iván Espinoza Ortiz**  
**Universidad de Santiago de Chile**  
**2022**

## **1- Introducción**

Un maestro contratista de interiores debe colocar azulejos a una línea horizontal de paredes de un cierto ancho, combinando azulejos negros y rojos.

Antes de comprar el material necesario, se requiere confeccionar un muestrario de opciones de combinación de los azulejos rojos y negros, según el ancho de la pared (en términos de la cantidad de azulejos que caben horizontalmente en ella).

Según criterio del cliente, no se ven bien los azulejos rojos contiguos, y por ello propone que cada vez que ocurra eso, se deberá colocar una aplicación dorada entre ellos, a modo de separación y sin afectar el ancho.

En este sentido, el maestro contratista debe generar el muestrario incluyendo todas las posibles combinaciones de azulejos rojos y negros para una pared con cierta capacidad horizontal de azulejos. Cada opción de dicho muestrario será revisada por el cliente para decidir el diseño que finalmente colocará en sus paredes.

Para el desarrollo de este problema, se necesita diseñar e implementar un algoritmo que dada una cantidad de azulejos, aplicando las características del problema, indique la cantidad de aplicaciones doradas necesarias para elaborar un muestrario de secuencias y las diferentes opciones analizadas.

Para la realización de este algoritmo se utilizará el lenguaje de programación C y lo aprendido en clases de Análisis y Estructura de algoritmos.

## **2-Solución propuesta**

El enfoque de esta solución se basa en la generación de números binarios.

La solución propuesta estará dividida en 4 funciones importantes.

Comenzando con la representación inicial del problema:

0= Negro

1= Rojo

a- Función Condición de término.

Función que se crea para darle una condición de término a la generación de combinaciones posibles de azulejos rojos y negros.

Esta condición es: Si todos los azulejos son rojos, termina la ejecución.

Pseudocódigo:

```
Condicion_termino( arreglo, largo)
for i<-0 to largo
  if arreglo[i]=0
    return(0)
return (1)
```

Complejidad:  $O(n)$

b- Función que crea todas las combinaciones posibles de azulejos.

Función que permite crear todas las combinaciones posibles de azulejos.

Se guía con una combinación de condiciones para cumplir este objetivo.

Las condiciones serían las siguientes:

-Para que termine todos los azulejos deben ser rojos.

-Si el ultimo azulejo es negro, se pone rojo

-Si el ultimo azulejo es rojo, anda a la penultima posición, retrocedes hasta ver un azulejo negro, ese azulejo negro se vuelve rojo, avanzas una posición y desde esa posición hacia el resto de azulejos de la derecha se ponen negros.

Pseudocódigo:

Combinacion(arreglo,largo):Combiarreglos

combiarreglos <-crearMatriz(largo)

auxAncho<-0

for j<-0 to largo

    combiarreglos[auxAncho][j]=arreglo[j]

while condiciontermino(arreglo,largo)=0

    if arreglo[largo-1]=0

        arreglo[largo-1]<-1

        for j<-0 to largo

            combiarreglos[auxAncho][j]=arreglo[j]

    else

        largoAux<-largo-2

        while arreglo[largoAux]=1

            largoAux<-largoAux-1

        arreglo[largoAux]<-1

        largoAux<-largoAux+1

        for i<-largoAux to largo

            arreglo[i]<-0

        for j<-0 to largo

            combiarreglos[auxAncho][j]<-arreglo[j]

return combiarreglos

Complejidad:  $O(n^2)$

c- Función que cuenta las aplicaciones doradas de los azulejos.

Función que permite contar todas las aplicaciones doradas que se van a aplicar en el muestrario, si encuentra dos posiciones que valen 1 de forma consecutiva, el contador sube en 1.

```
Doradas(listaArreglos,largo): totalDoradas
cantCombi<-Combinaciones(largo)
totalDoradas<-0
for i<-0 to cantCombi
  for j<-0 to largo-1
    if listaArreglos[i][j]=1
      if listaArreglos[i][j+1]=1
        totalDoradas<-totalDoradas+1

Return(totalDoradas)
```

Complejidad:  $O(n^2)$

d- Función que escribe la representación completa del muestrario de opciones.

Primero escribe todas las filas con R D y después imprime todas las últimas posiciones de cada fila.

Pseudocódigo:

```
Escribir(listaArreglos,largo,nombreambrivo,cantDoradas):textoSalida
TextoSalida<-fopen(nombreambrivo)
cantCombi<-combinaciones(largo)
write(textosalida,cantDoradas,cantCombi)
for i<-0 to cantCombi
  for <-0 to largo-1
    if listaArreglos[i][j]=1
      if listaArreglos[i][j+1]=1
        escribir(textosalida,R D)
      else
        escribir(textosalida, R)
    else
      escribir(textosalida,N)
  if listaArreglos[i][largo-1]=1
    escribir(textosalida,R)
  else
    escribir(textosalida,N)
  cerrararchivo(textosalida)
```

Complejidad :  $O(n^2)$

### 3-Resultados y análisis

Tras hacer análisis y pruebas de los distintos resultados

Las combinaciones posibles aumentan de forma exponencial mientras más largo es el tamaño de los azulejos.

En el equipo solo se pudo ejecutar hasta un azulejo de 24 combinaciones.

salida	C:\Users\espin\Desktop\Semestre 02-202...	07-09-2022 15:13	51.713 KB	Archivo OUT
salida	C:\Users\espin\Desktop\Semestre 02-202...	07-09-2022 14:52	108.545 KB	Archivo OUT
salida	C:\Users\espin\Desktop\Semestre 02-202...	07-09-2022 14:53	227.329 KB	Archivo OUT
salida	C:\Users\espin\Desktop\Semestre 02-202...	07-09-2022 15:15	991.233 KB	Archivo OUT
salida	C:\Users\espin\Desktop\Semestre 02-202...	07-09-2022 14:54	475.137 KB	Archivo OUT
salida	C:\Users\espin\Desktop\Semestre 02-202...	07-09-2022 14:49	51.713 KB	Archivo OUT
salida	C:\Users\espin\Desktop\Semestre 02-202...	07-09-2022 14:44	1 KB	Archivo OUT

Figura 1: peso de distintos archivos de salida.

Archivo Edición Formato Ver Ay  
12 16

Figura 2: Cantidad de aplicaciones y combinaciones en un archivo de 4 azulejos.

Peso: 1kb

Tiempo de ejecución: 0.0

```
clock_t : 0.0000000000000000
```

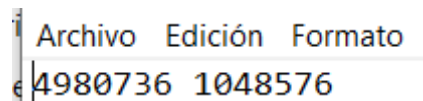
Archivo Edición Fori  
2304 1024

Figura 3: Cantidad de aplicaciones y Combinaciones en un archivo de 10 azulejos

Peso: 26kb

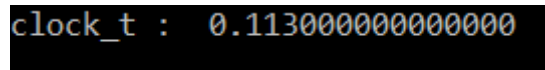
Tiempo de ejecución: 0.0

```
clock_t : 0.0000000000000000
```

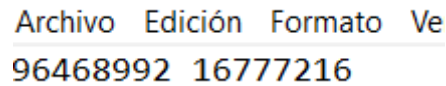


Archivo Edición Formato  
4980736 1048576

**Figura 4: Cantidad de aplicaciones y combinaciones en un archivo de 20 azulejos.**  
**Peso: 51.713kb**



clock\_t : 0.113000000000000



Archivo Edición Formato Ve  
96468992 16777216

**Figura 4: Cantidad de Aplicaciones y Combinaciones en un archivo de 24 azulejos.**  
**Peso: 991.233KB**  
**Tiempo de ejecución: aproximado 10min**

#### 4-Conclusiones

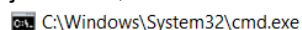
En conclusión, el objetivo de la tarea se cumplió exitosamente, logrando la solución esperada. Respecto a los resultados anteriores, se cumplió lo esperado.

Una de las ventajas de esta solución es que está implementada por llamadas de funciones y lo hace un programa bien estructurado.

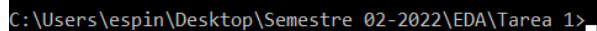
Una de las posibles desventajas y mejoras del trabajo, podría existir una forma de realizar el algoritmo de forma recursiva para un menor tiempo de ejecución esperado.

#### Anexo manual de usuario

Para ejecutar correctamente el programa, se debe escribir vía línea de comando el nombre del ejecutable, la cantidad de azulejos y el nombre del archivo de salida(en su defecto Salida.out).



C:\Windows\System32\cmd.exe



C:\Users\espín\Desktop\Semestre 02-2022\EDA\Tarea 1>

**Figura 5: Línea de comando.**

Ejemplo:

**Nombre Programa 4 Salida.out**