ALGORITMIA

PRÁCTICA 4

Héctor Lavandeira Fernández
UO277303 | UNIVERSIDAD DE OVIEDO – CURSO 2021/22

Características principales del ordenador:

Procesador:	i5-8250U
Memoria RAM:	8GB

Explica brevemente cómo funciona tu algoritmo

Para empezar, tengo un método principal *coloracionMapa()* que itera por cada país, y le da el color correspondiente:

```
public void coloracionMapa() {
    String[] paisesNombres = paises.keySet().toArray(new String[paises.size()]);

    añadirColor(paisesNombres[0], listaColores.get(0));
    for (int i = 1; i < paises.size(); i++) {
        añadirColor(paisesNombres[i], getColorPosible(paisesNombres[i]));
    }
}
```

En el método *añadirColor()*, añado el país con su color correspondiente al HashMap de colores, y añado el color a la lista de colores usados, en caso de que este no haya sido añadido ya.

```
private void añadirColor(String pais, String color) {
    colores.put(pais, color);
    if (!coloresUsados.contains(color)) {
        coloresUsados.add(color);
    }
}
```

El método *getColorPosible()* utiliza un ArrayList donde guardará los colores de las fronteras (si tienen) del país que se introduce por parámetro. Primero itera por cada frontera del país, y añade a la lista su color.

Después, itera sobre la lista de colores y devuelve el primero que no ha sido añadido a la lista de colores de las fronteras del país, así me aseguro de que utiliza el color con más prioridad posible.

```
private String getColorPosible(String pais) {
    ArrayList<String> coloresFronteras = new ArrayList<String>();
    for (int i = 0; i < paises.get(pais).size(); i++) {
        if (colores.get(paises.get(pais).get(i)) != null) {
            if (!coloresFronteras.contains(colores.get(paises.get(pais).get(i)))) {
                coloresFronteras.add(colores.get(paises.get(pais).get(i)));
            }
        }
    }

    for (int i = 0; i < listaColores.size(); i++) {
        if (!coloresFronteras.contains(listaColores.get(i))) {
            return listaColores.get(i);
        }
    }
    return "";
}
```

¿Cuántos colores has necesitado para resolver el problema dado?

He necesitado un total de 5 colores:

```
COLORES:
Se han utilizado 5 colores:
        Rojo
        Azul
        Verde
        Amarillo
        Negro
```

¿Podría cambiar el número de colores necesario si utilizas un orden diferente?

Sí, podría cambiar el número de colores dependiendo del orden de los países.

¿Cuántos colores utilizarías, como mucho, en una solución óptima?

En una solución óptima, si se encuentra el orden necesario de los países, podrían utilizarse 4, e incluso 3 colores.

¿Cuál es la complejidad temporal de tu algoritmo? Explícala brevemente

Para empezar, el método *añadirColor()* tiene una complejidad O(1), ya que solo contiene instrucciones simples.

El método *getColorPosible()* tiene dos bucles. El primero itera sobre todas las fronteras de un país, y en su interior solo hay instrucciones simples, por lo que su complejidad es O(n). El segundo bucle itera sobre todos los colores, por lo que su complejidad sería O(n). Por tanto, la complejidad del método completo es $O_{bucle\ 1} + O_{bucle\ 2} = O(n) + O(n) = O(n)$.

Por último, el método *coloracionMapa()* itera sobre todos los países (tamaño n del problema), y en su interior realiza una llamada al método *añadirColor()* y otra a *getColorPosible()*, por tanto, su complejidad es $O(n) * O(n) = O(n^2)$.