Ejercicio resuelto en clase en semana 3

Ejercicio 1 (Punto de encuentro)**.** Consideramos un juego en el que participan dos jugadores, A y B, que se desarrolla sobre un tablero de $M \times M$ casillas, organizadas en M filas y M columnas, numeradas de 1 a M. Una *posición* en el tablero es un par (f, c), donde f es un número de fila, $1 \leq f \leq M$, y c es un número de columna, $1 \leq c \leq M$. El conjunto de todas las posiciones del tablero es $T = \{1, 2, \dots, M\}^2$.

Cada jugador tiene un ficha, que puede desplazar por el tablero realizando movimientos que llevan su ficha desde a una casilla a otra adyacente, considerando como casillas *adyacentes* a las que están inmediatamente arriba, abajo, a la izquierda o a la derecha. Específicamente una casilla en posición (f, c) es adyacente a las casillas en las posiciones del conjunto

$$Ad(f, c) = \Big\{(f-1, c), (f+1, c), (f, c-1), (f, c+1)\Big\} \bigcap T\,.$$

Cada casilla del tablero está coloreada según un código que establece restricciones de movimiento para los jugadores:

- Verde: habilitada para ambos jugadores.
- Rojo: inhabilitada para ambos jugadores.
- Azul: habilitada exclusivamente para A.
- Blanco: habilitada exclusivamente para B.

Al inicio del juego las fichas de A y B están en ciertas casillas iniciales, en posiciones p_A y p_B, respectivamente. Asumimos que la casilla inicial de cada jugador tiene un color que lo habilita a estar allí. El objetivo del juego (colaborativo) es que A y B desplacen sus fichas hasta encontrarse en alguna casilla en común, realizando la menor cantidad posible de movimientos en total. Tanto los movimientos realizados como la posición del punto de encuentro, p_C, debe ser determinada por los jugadores procurando minimizar la cantidad total de movimientos. Los jugadores A y B no tienen por qué mover alternadamente sus fichas; por ejemplo A podría llegar a p_C en menos movimientos que B o viceversa. En la figura 1.1 se muestra un tablero de ejemplo.

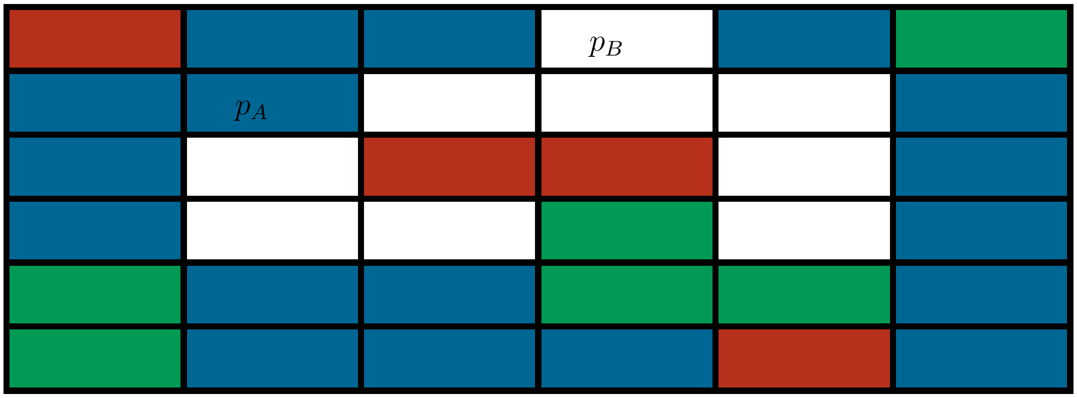

Figura 1.1: Ejemplo de tablero.

Queremos construir un algoritmo para resolver este problema. Definimos un *recorrido válido* para un jugador como una secuencia de posiciones de casillas habilitadas para ese jugador, $p_1, p_2, \ldots, p_r$, tal que las casillas en posiciones p_i, p_{i+1} son adyacentes para todo i, $1 \leq i < r$. Definimos el *largo* de este recorrido como la cantidad de movimientos realizados, $r - 1$. Nuestro algoritmo toma como entrada un tablero, representado como una matriz de colores, y las posiciones iniciales p_A y p_B; como salida genera un recorrido válido para A que comienza en p_A y un recorrido válido para B que comienza en p_B, tales que la suma de los largos de estos recorridos es la menor posible, sujeto a que ambos recorridos terminan en una misma posición, p_C. Si no existen tales recorridos nuestro algoritmo debe informar que no existe solución.

(a) Dé un algoritmo para resolver el problema planteado. Su algoritmo debe admitir una implementación cuyo tiempo de ejecución es $O(M^2)$.

(b) Demuestre la corrección de su algoritmo.

(c) Demuestre que su algoritmo admite una implementación cuyo tiempo de ejecución es $O(M^2)$.