

Descripción

Estás atrapado en un sueño del cuál te cuesta despertar. En este sueño tan bizarro, te encuentras en un tablero de ajedrez de 1000×1000 . Lo curioso es que estás subido encima de la pieza del caballo en la casilla (x_1, y_1) . Por algún motivo, existe una casilla especial (x_2, y_2) donde se encuentra la pieza del rey contrincante. Tú sabes que esa pieza no se puede mover y tu meta es apoderarte de esa casilla llegando a ella para derrocar al rey contrincante. Es la única forma que puedes despertar de esta pesadilla. Dicho esto, ahora te preguntas cuál será el mínimo número de pasos que tú y tu caballo deben dar para llegar hacia la pieza del rey contrincante.

Entrada

2 líneas con 2 números enteros cada uno. La primera línea tiene dos números enteros (x_1, y_1) , separados por un espacio, que representan la casilla donde empiezas.

La segunda línea tiene dos números enteros (x_2, y_2) , separados por un espacio, que representa la casilla del rey contrincante.

Salida

Un solo número, que representa la mínima cantidad de movimientos que deben dar tú y tu caballo para llegar desde la casilla del inicio hasta la del rey contrincante.

Notas

- Las posiciones comienzan en 0, es decir, $(0, 0)$ representa la esquina inferior izquierda del tablero.
- El caballo de ajedrez se mueve en forma de L , es decir, desde una casilla con posición (x, y) puede moverse a 8 posibles casillas, con coordenadas $(x \pm 1, y \pm 2)$ o $(x \pm 2, y \pm 1)$ (siempre y cuando la casilla esté dentro del tablero).

Solución:

Este problema es un ejemplo clásico del uso del algoritmo de búsqueda en amplitud (BFS). La idea principal es desarrollar la búsqueda en amplitud guardando la distancia desde la casilla inicial hacia cada casilla. Empezamos desde la casilla inicial, y realizamos el recorrido en amplitud considerando casillas adyacentes válidas, las casillas a las que se puede llegar con un movimiento en L y están dentro del tablero. Al llegar por primera vez a la casilla deseada, imprimimos la distancia calculada y terminamos el algoritmo.