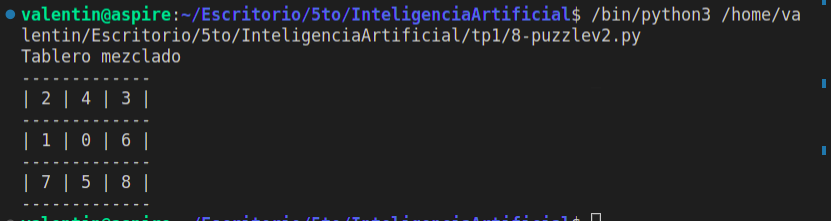
**Trabajo Práctico 8 Puzzle**

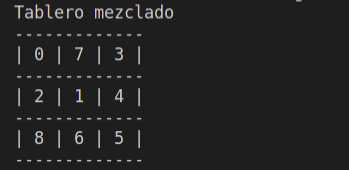
Valentín Faraz

Ing Computación

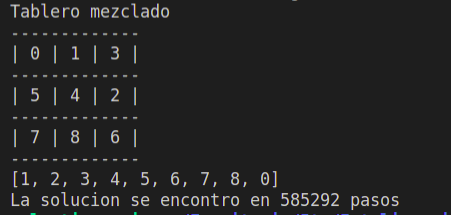
5to año

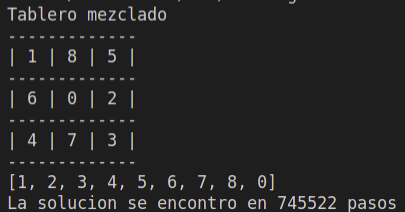
**Mezclar 50 veces random**



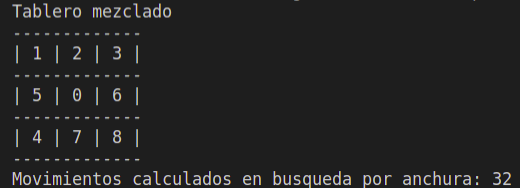


**Búsqueda random**

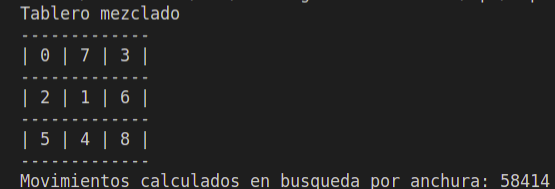




**Búsqueda por anchura**

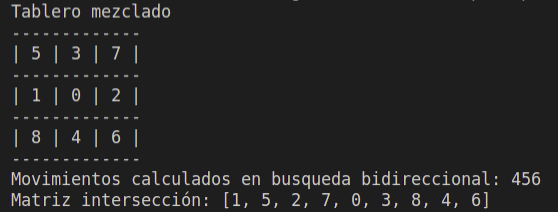


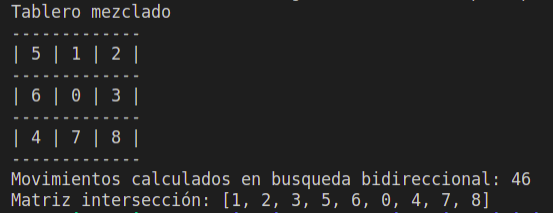
La solución de la matriz anterior se encontró en el nivel 4



El movimiento anterior se encontró en el nivel 38

**Búsqueda bidireccional**





**Código**

**import random**

**arbol=[]**

**arbol\_b=[]**

**def info\_tablero(tablero):**

**pos\_moviles=[]**

**for n in range(len(tablero)):**

**if n in [0,1,3,4,6,7]:**

**#Se puede mover hacia la derecha**

**if tablero[n+1] == 0:**

**pos\_moviles.append(n)**

**if n in [0,1,2,3,4,5]:**

**#Se puede mover hacia abiajo**

**if tablero[n+3]==0:**

**pos\_moviles.append(n)**

**if n in [1,2,4,5,7,8]:**

**#Se puede mover hacia la izquierda**

**if tablero[n-1] == 0:**

**pos\_moviles.append(n)**

**if n in [3,4,5,6,7,8]:**

**#Se puede mover hacia arriba**

**if tablero[n-3]==0:**

**pos\_moviles.append(n)**

**return(pos\_moviles)**

**def mostrar(tablero):**

**print('-------------')**

**print('|',tablero[0],'|',tablero[1],'|',tablero[2],'|')**

**print('-------------')**

**print('|',tablero[3],'|',tablero[4],'|',tablero[5],'|')**

**print('-------------')**

**print('|',tablero[6],'|',tablero[7],'|',tablero[8],'|')**

**print('-------------')**

**def mezclar(tablero\_m):**

**movimientos=50**

**while movimientos > 0:**

**movimientos=movimientos-1**

**pos\_moviles = info\_tablero(tablero\_m)**

**ran = random.choice(pos\_moviles)**

**indice\_vacio = tablero\_m.index(0)**

**tablero\_m[indice\_vacio]=tablero\_m[ran]**

**tablero\_m[ran]=0**

**return tablero\_m**

**def busqueda\_random(tablero):**

**pasos=0**

**while tablero != [1,2,3,4,5,6,7,8,0]:**

**pasos=pasos+1**

**pos\_moviles = info\_tablero(tablero)**

**ran = random.choice(pos\_moviles)**

**indice\_vacio = tablero.index(0)**

**tablero[indice\_vacio]=tablero[ran]**

**tablero[ran]=0**

**print(tablero)**

**print("La solucion se encontro en",pasos,"pasos")**

**return pasos**

**def busqueda\_anchura(tablero):**

**aux=tablero.copy()**

**pos\_moviles = info\_tablero(tablero)**

**indice\_vacio = tablero.index(0)**

**desarrollo\_tablero=[]**

**for n in pos\_moviles:**

**tablero=aux.copy()**

**tablero[indice\_vacio]=tablero[n]**

**tablero[n]=0**

**desarrollo\_tablero.append(tablero)**

**arbol.append(desarrollo\_tablero)**

**def busqueda\_bidireccional(tablero):**

**aux=tablero.copy()**

**pos\_moviles = info\_tablero(tablero)**

**indice\_vacio = tablero.index(0)**

**desarrollo\_tablero=[]**

**for n in pos\_moviles:**

**tablero=aux.copy()**

**tablero[indice\_vacio]=tablero[n]**

**tablero[n]=0**

**desarrollo\_tablero.append(tablero)**

**arbol\_b.append(desarrollo\_tablero)**

**if \_\_name\_\_ == "\_\_main\_\_":**

**tablero=[1,2,3,**

**4,5,6,**

**7,8,0]**

**tablero\_solucion=[1,2,3,**

**4,5,6,**

**7,8,0]**

**#Mezclar el tablero**

**tablero\_mezclado = mezclar(tablero)**

**print("Tablero mezclado")**

**mostrar(tablero\_mezclado)**

**#Buscar random**

**#Encontro la solucion una vez en el paso n° 190.245 y otra vez en el paso n° 1.738.696**

**busqueda\_random(tablero\_mezclado)**

**#Busqueda en anchura**

**busqueda\_anchura(tablero\_mezclado)**

**tabla=[]**

**numero\_nivel=0**

**iteracion=0**

**for nivel in arbol:**

**#print('arbol:',iteracion,arbol)**

**for tabla in nivel:**

**#print (tabla)**

**busqueda\_anchura(tabla)**

**if tabla==tablero\_solucion:**

**movimientos = arbol.index(nivel)**

**print('Movimientos calculados en busqueda por anchura:',movimientos)**

**quit()**

**#Busqueda bidireccional**

**busqueda\_bidireccional(tablero\_mezclado)**

**busqueda\_anchura(tablero\_solucion)**

**tabla=[]**

**numero\_nivel=0**

**iteracion=0**

**for nivel,nivel\_b in zip(arbol, arbol\_b):**

**for tabla,tabla\_b in zip(nivel,nivel\_b):**

**for validacion in arbol:**

**for matriz in validacion:**

**if matriz==tabla\_b:**

**movimientos = arbol.index(validacion)**

**print('Movimientos calculados en busqueda bidireccional:',movimientos)**

**print("Matriz intersección:",tabla\_b)**

**quit()**

**busqueda\_bidireccional(tabla\_b)**

**busqueda\_anchura(tabla)**