Paolo_Valerio_Testa_AG1

September 21, 2024

Actividad Guiada 1 de Algoritmos de Optimizacion

Nombre: Paolo Valerio Testa

https://colab.research.google.com/drive/1EyG5Zz4dlI8-c-wKuivZMwG-7goYYC9I?usp=sharing

https://github.com/pvt198/03MAIR-Algoritmos-de-Optimizaci-n.git

[1]:
```
#Torres de Hanoi - Divide y venceras
#############################################################################



#####################################################################
def Torres_Hanoi(N, desde, hasta):
  #N - Nº de fichas
  #desde - torre inicial
  #hasta - torre fina
  if N==1 :
    print("Lleva la ficha desde " + str(desde) + " hasta " + str(hasta))

  else:
    Torres_Hanoi(N-1, desde, 6-desde-hasta)
    print("Lleva la ficha desde " + str(desde) + " hasta " + str(hasta))
    Torres_Hanoi(N-1, 6-desde-hasta,  hasta)

Torres_Hanoi(3, 1, 3)
#####################################################################
```

```
Lleva la ficha desde 1 hasta 3
Lleva la ficha desde 1 hasta 2
Lleva la ficha desde 3 hasta 2
Lleva la ficha desde 1 hasta 3
Lleva la ficha desde 2 hasta 1
Lleva la ficha desde 2 hasta 3
Lleva la ficha desde 1 hasta 3
```

[5]:
```
def Torres_Hanoi(N, desde, hasta, torres):
    # Caso base: si solo hay una ficha, moverla directamente
    if N == 1:
        # Quitar la ficha de la torre "desde" y agregarla a la torre "hasta"
```

```
        disk = torres[desde].pop()  # Extraer la ficha de la torre "desde"
        torres[hasta].append(disk)  # Colocar la ficha en la torre "hasta"

        # Imprimir el movimiento realizado
        print(f"Lleva la ficha {disk} desde torre {desde+1} hasta torre␣
 ↪{hasta+1}")
        # Mostrar el estado actual de las torres
        print("Estado de las torres:")
        mostrar_torres(torres)
    else:
        # Mover las N-1 fichas superiores a la torre auxiliar
        Torres_Hanoi(N-1, desde, 3-desde-hasta, torres)

        # Mover la ficha N a la torre de destino
        disk = torres[desde].pop()  # Extraer la ficha de la torre "desde"
        torres[hasta].append(disk)  # Colocar la ficha en la torre "hasta"
        print(f"Lleva la ficha {disk} desde torre {desde+1} hasta torre␣
 ↪{hasta+1}")
        print("Estado de las torres:")
        mostrar_torres(torres)

        # Mover las N-1 fichas de la torre auxiliar a la torre de destino
        Torres_Hanoi(N-1, 3-desde-hasta, hasta, torres)

# Función para mostrar el estado de las torres
def mostrar_torres(torres):
    for i, torre in enumerate(torres):
        # Imprimir el número de la torre y las fichas que contiene
        print(f"Torre {i+1}: {torre}")
    # Separador visual entre cada movimiento
    print("-" * 20)

# Inicializamos las 3 torres con N discos
N = 3
torres = [list(range(N, 0, -1)), [], []]  # Torre 1 tiene N fichas, las demás␣
 ↪están vacías

# Mostrar estado inicial de las torres
print("Estado inicial de las torres:")
mostrar_torres(torres)

# Ejecutar el algoritmo de Torres de Hanói
Torres_Hanoi(N, 0, 2, torres)
```

```
Estado inicial de las torres:
Torre 1: [3, 2, 1]
Torre 2: []
```

```
Torre 3: []
--------------------
Lleva la ficha 1 desde torre 1 hasta torre 3
Estado de las torres:
Torre 1: [3, 2]
Torre 2: []
Torre 3: [1]
--------------------
Lleva la ficha 2 desde torre 1 hasta torre 2
Estado de las torres:
Torre 1: [3]
Torre 2: [2]
Torre 3: [1]
--------------------
Lleva la ficha 1 desde torre 3 hasta torre 2
Estado de las torres:
Torre 1: [3]
Torre 2: [2, 1]
Torre 3: []
--------------------
Lleva la ficha 3 desde torre 1 hasta torre 3
Estado de las torres:
Torre 1: []
Torre 2: [2, 1]
Torre 3: [3]
--------------------
Lleva la ficha 1 desde torre 2 hasta torre 1
Estado de las torres:
Torre 1: [1]
Torre 2: [2]
Torre 3: [3]
--------------------
Lleva la ficha 2 desde torre 2 hasta torre 3
Estado de las torres:
Torre 1: [1]
Torre 2: []
Torre 3: [3, 2]
--------------------
Lleva la ficha 1 desde torre 1 hasta torre 3
Estado de las torres:
Torre 1: []
Torre 2: []
Torre 3: [3, 2, 1]
--------------------
```

[ ]:

[6]:
```
#Cambio de monedas - Técnica voraz
#############################################################################
SISTEMA = [11, 5 , 3, 1  ]
#####################################################################
def cambio_monedas(CANTIDAD,SISTEMA):
#....
  SOLUCION = [0]*len(SISTEMA)
  ValorAcumulado = 0

  for i,valor in enumerate(SISTEMA):
    monedas = (CANTIDAD-ValorAcumulado)//valor
    SOLUCION[i] = monedas
    ValorAcumulado = ValorAcumulado + monedas*valor

    if CANTIDAD == ValorAcumulado:
      return SOLUCION

  print("No es posible encontrar solucion")

cambio_monedas(25,SISTEMA)



#####################################################################
```

[6]:
```
[2, 0, 1, 0]
```

[9]:
```
#Cambio de monedas - Técnica voraz
# SISTEMA DE QUE NO SE PUEDE ENCUENTRAR SOLUCION!
#############################################################################
SISTEMA = [7, 5 , 3,  ]
#####################################################################
def cambio_monedas(CANTIDAD,SISTEMA):
#....
  SOLUCION = [0]*len(SISTEMA)
  ValorAcumulado = 0

  for i,valor in enumerate(SISTEMA):
    monedas = (CANTIDAD-ValorAcumulado)//valor
    SOLUCION[i] = monedas
    ValorAcumulado = ValorAcumulado + monedas*valor

    if CANTIDAD == ValorAcumulado:
      return SOLUCION

  print("No es posible encontrar solucion")

cambio_monedas(25,SISTEMA)
```

```
No es posible encontrar solucion
```

[ ]:

[14]:
```python
#N Reinas - Vuelta Atrás()
#############################################################################

#Verifica que en la solución parcial no hay amenzas entre reinas
#####################################################################
def es_prometedora(SOLUCION,etapa):
#####################################################################
  #print(SOLUCION)
  #Si la solución tiene dos valores iguales no es valida => Dos reinas en la␣
 ↪misma fila
  for i in range(etapa+1):
    #print("El valor " + str(SOLUCION[i]) + " está " +  str(SOLUCION.
 ↪count(SOLUCION[i])) + " veces")
    if SOLUCION.count(SOLUCION[i]) > 1:
      return False

    #Verifica las diagonales
    for j in range(i+1, etapa +1 ):
      #print("Comprobando diagonal de " + str(i) + " y " + str(j))
      if abs(i-j) == abs(SOLUCION[i]-SOLUCION[j]) : return False
  return True

#Traduce la solución al tablero
#####################################################################
def escribe_solucion(S):
#####################################################################
  n = len(S)
  for x in range(n):
    print("")
    for i in range(n):
      if S[i] == x+1:
        print(" X " , end="")
      else:
        print(" - ", end="")



#Proceso principal de N-Reinas
#####################################################################
def reinas(N, solucion=[],etapa=0):
#####################################################################
### ....
  if len(solucion) == 0:         # [0,0,0...]
```

```
    solucion = [0 for i in range(N) ]

  for i in range(1, N+1):
    solucion[etapa] = i
    if es_prometedora(solucion, etapa):
      if etapa == N-1:
        print(solucion)
        escribe_solucion(solucion)
      else:
        reinas(N, solucion, etapa+1)
    else:
      None

  solucion[etapa] = 0

reinas(8,solucion=[],etapa=0)
```

```
[1, 5, 8, 6, 3, 7, 2, 4]

X  -  -  -  -  -  -  -
-  -  -  -  -  -  X  -
-  -  -  -  X  -  -  -
-  -  -  -  -  -  -  X
-  X  -  -  -  -  -  -
-  -  -  X  -  -  -  -
-  -  -  -  -  X  -  -
-  -  X  -  -  -  -  - [1, 6, 8, 3, 7, 4, 2, 5]

X  -  -  -  -  -  -  -
-  -  -  -  -  -  X  -
-  -  -  X  -  -  -  -
-  -  -  -  -  X  -  -
-  -  -  -  -  -  -  X
-  X  -  -  -  -  -  -
-  -  -  -  X  -  -  -
-  -  X  -  -  -  -  - [1, 7, 4, 6, 8, 2, 5, 3]

X  -  -  -  -  -  -  -
-  -  -  -  -  X  -  -
-  -  -  -  -  -  -  X
-  -  X  -  -  -  -  -
-  -  -  -  -  -  X  -
-  -  -  X  -  -  -  -
-  X  -  -  -  -  -  -
-  -  -  -  X  -  -  - [1, 7, 5, 8, 2, 4, 6, 3]

X  -  -  -  -  -  -  -
-  -  -  -  X  -  -  -
```

```
-  -  -  -  -  -  -  X
-  -  -  -  -  X  -  -
-  -  X  -  -  -  -  -
-  -  -  -  -  -  X  -
-  X  -  -  -  -  -  -
-  -  -  X  -  -  -  - [2, 4, 6, 8, 3, 1, 7, 5]

-  -  -  -  -  X  -  -
X  -  -  -  -  -  -  -
-  -  -  -  X  -  -  -
-  X  -  -  -  -  -  -
-  -  -  -  -  -  -  X
-  -  X  -  -  -  -  -
-  -  -  -  -  -  X  -
-  -  -  X  -  -  -  - [2, 5, 7, 1, 3, 8, 6, 4]

-  -  -  X  -  -  -  -
X  -  -  -  -  -  -  -
-  -  -  -  X  -  -  -
-  -  -  -  -  -  -  X
-  X  -  -  -  -  -  -
-  -  -  -  -  -  X  -
-  -  X  -  -  -  -  -
-  -  -  -  -  X  -  - [2, 5, 7, 4, 1, 8, 6, 3]

-  -  -  -  X  -  -  -
X  -  -  -  -  -  -  -
-  -  -  -  -  -  -  X
-  -  -  X  -  -  -  -
-  X  -  -  -  -  -  -
-  -  -  -  -  -  X  -
-  -  X  -  -  -  -  -
-  -  -  -  -  X  -  - [2, 6, 1, 7, 4, 8, 3, 5]

-  -  X  -  -  -  -  -
X  -  -  -  -  -  -  -
-  -  -  -  -  -  X  -
-  -  -  -  X  -  -  -
-  -  -  -  -  -  -  X
-  X  -  -  -  -  -  -
-  -  -  X  -  -  -  -
-  -  -  -  -  X  -  - [2, 6, 8, 3, 1, 4, 7, 5]

-  -  -  -  X  -  -  -
X  -  -  -  -  -  -  -
-  -  -  X  -  -  -  -
-  -  -  -  -  X  -  -
-  -  -  -  -  -  -  X
```

```
-  X  -  -  -  -  -  -
-  -  -  -  -  -  X  -
-  -  X  -  -  -  -  - [2, 7, 3, 6, 8, 5, 1, 4]

-  -  -  -  -  -  X  -
X  -  -  -  -  -  -  -
-  -  X  -  -  -  -  -
-  -  -  -  -  -  -  X
-  -  -  -  -  X  -  -
-  -  -  X  -  -  -  -
-  X  -  -  -  -  -  -
-  -  -  -  X  -  -  - [2, 7, 5, 8, 1, 4, 6, 3]

-  -  -  -  X  -  -  -
X  -  -  -  -  -  -  -
-  -  -  -  -  -  -  X
-  -  -  -  -  X  -  -
-  -  X  -  -  -  -  -
-  -  -  -  -  -  X  -
-  X  -  -  -  -  -  -
-  -  -  X  -  -  -  - [2, 8, 6, 1, 3, 5, 7, 4]

-  -  -  X  -  -  -  -
X  -  -  -  -  -  -  -
-  -  -  -  X  -  -  -
-  -  -  -  -  -  -  X
-  -  -  -  -  X  -  -
-  -  X  -  -  -  -  -
-  -  -  -  -  -  X  -
-  X  -  -  -  -  -  - [3, 1, 7, 5, 8, 2, 4, 6]

-  X  -  -  -  -  -  -
-  -  -  -  -  X  -  -
X  -  -  -  -  -  -  -
-  -  -  -  -  -  X  -
-  -  -  X  -  -  -  -
-  -  -  -  -  -  -  X
-  -  X  -  -  -  -  -
-  -  -  -  X  -  -  - [3, 5, 2, 8, 1, 7, 4, 6]

-  -  -  -  X  -  -  -
-  -  X  -  -  -  -  -
X  -  -  -  -  -  -  -
-  -  -  -  -  -  X  -
-  X  -  -  -  -  -  -
-  -  -  -  -  -  -  X
-  -  -  -  -  X  -  -
-  -  -  X  -  -  -  - [3, 5, 2, 8, 6, 4, 7, 1]
```

```
-  -  -  -  -  -  -  X
-  -  X  -  -  -  -  -
X  -  -  -  -  -  -  -
-  -  -  -  -  X  -  -
-  X  -  -  -  -  -  -
-  -  -  -  X  -  -  -
-  -  -  -  -  -  X  -
-  -  -  X  -  -  -  - [3, 5, 7, 1, 4, 2, 8, 6]

-  -  -  X  -  -  -  -
-  -  -  -  -  X  -  -
X  -  -  -  -  -  -  -
-  -  -  -  X  -  -  -
-  X  -  -  -  -  -  -
-  -  -  -  -  -  -  X
-  -  X  -  -  -  -  -
-  -  -  -  -  -  X  - [3, 5, 8, 4, 1, 7, 2, 6]

-  -  -  -  X  -  -  -
-  -  -  -  -  -  X  -
X  -  -  -  -  -  -  -
-  -  -  X  -  -  -  -
-  X  -  -  -  -  -  -
-  -  -  -  -  -  -  X
-  -  -  -  -  X  -  -
-  -  X  -  -  -  -  - [3, 6, 2, 5, 8, 1, 7, 4]

-  -  -  -  -  X  -  -
-  -  X  -  -  -  -  -
X  -  -  -  -  -  -  -
-  -  -  -  -  -  -  X
-  -  -  X  -  -  -  -
-  X  -  -  -  -  -  -
-  -  -  -  -  -  X  -
-  -  -  -  X  -  -  - [3, 6, 2, 7, 1, 4, 8, 5]

-  -  -  -  X  -  -  -
-  -  X  -  -  -  -  -
X  -  -  -  -  -  -  -
-  -  -  -  -  X  -  -
-  -  -  -  -  -  -  X
-  X  -  -  -  -  -  -
-  -  -  X  -  -  -  -
-  -  -  -  -  -  X  - [3, 6, 2, 7, 5, 1, 8, 4]

-  -  -  -  -  X  -  -
-  -  X  -  -  -  -  -
```

```
X  -  -  -  -  -  -  -
-  -  -  -  -  -  -  X
-  -  -  -  X  -  -  -
-  X  -  -  -  -  -  -
-  -  -  X  -  -  -  -
-  -  -  -  -  -  X  - [3, 6, 4, 1, 8, 5, 7, 2]

-  -  -  X  -  -  -  -
-  -  -  -  -  -  -  X
X  -  -  -  -  -  -  -
-  -  X  -  -  -  -  -
-  -  -  -  -  X  -  -
-  X  -  -  -  -  -  -
-  -  -  -  -  -  X  -
-  -  -  -  X  -  -  - [3, 6, 4, 2, 8, 5, 7, 1]

-  -  -  -  -  -  -  X
-  -  -  X  -  -  -  -
X  -  -  -  -  -  -  -
-  -  X  -  -  -  -  -
-  -  -  -  -  X  -  -
-  X  -  -  -  -  -  -
-  -  -  -  -  -  X  -
-  -  -  -  X  -  -  - [3, 6, 8, 1, 4, 7, 5, 2]

-  -  -  X  -  -  -  -
-  -  -  -  -  -  -  X
X  -  -  -  -  -  -  -
-  -  -  -  X  -  -  -
-  -  -  -  -  -  X  -
-  X  -  -  -  -  -  -
-  -  -  -  -  X  -  -
-  -  X  -  -  -  -  - [3, 6, 8, 1, 5, 7, 2, 4]

-  -  -  X  -  -  -  -
-  -  -  -  -  -  X  -
X  -  -  -  -  -  -  -
-  -  -  -  -  -  -  X
-  -  -  -  X  -  -  -
-  X  -  -  -  -  -  -
-  -  -  -  -  X  -  -
-  -  X  -  -  -  -  - [3, 6, 8, 2, 4, 1, 7, 5]

-  -  -  -  -  X  -  -
-  -  -  X  -  -  -  -
X  -  -  -  -  -  -  -
-  -  -  -  X  -  -  -
-  -  -  -  -  -  -  X
```

```
-  X  -  -  -  -  -  -
-  -  -  -  -  -  X  -
-  -  X  -  -  -  -  - [3, 7, 2, 8, 5, 1, 4, 6]

-  -  -  -  -  X  -  -
-  -  X  -  -  -  -  -
X  -  -  -  -  -  -  -
-  -  -  -  -  -  X  -
-  -  -  -  X  -  -  -
-  -  -  -  -  -  -  X
-  X  -  -  -  -  -  -
-  -  -  X  -  -  -  - [3, 7, 2, 8, 6, 4, 1, 5]

-  -  -  -  -  -  X  -
-  -  X  -  -  -  -  -
X  -  -  -  -  -  -  -
-  -  -  -  -  X  -  -
-  -  -  -  -  -  -  X
-  -  -  -  X  -  -  -
-  X  -  -  -  -  -  -
-  -  -  X  -  -  -  - [3, 8, 4, 7, 1, 6, 2, 5]

-  -  -  -  X  -  -  -
-  -  -  -  -  -  X  -
X  -  -  -  -  -  -  -
-  -  X  -  -  -  -  -
-  -  -  -  -  -  -  X
-  -  -  -  -  X  -  -
-  -  -  X  -  -  -  -
-  X  -  -  -  -  -  - [4, 1, 5, 8, 2, 7, 3, 6]

-  X  -  -  -  -  -  -
-  -  -  -  X  -  -  -
-  -  -  -  -  -  X  -
X  -  -  -  -  -  -  -
-  -  X  -  -  -  -  -
-  -  -  -  -  -  -  X
-  -  -  -  -  X  -  -
-  -  -  X  -  -  -  - [4, 1, 5, 8, 6, 3, 7, 2]

-  X  -  -  -  -  -  -
-  -  -  -  -  -  -  X
-  -  -  -  -  X  -  -
X  -  -  -  -  -  -  -
-  -  X  -  -  -  -  -
-  -  -  -  X  -  -  -
-  -  -  -  -  -  X  -
-  -  -  X  -  -  -  - [4, 2, 5, 8, 6, 1, 3, 7]
```

```
-  -  -  -  -  X  -  -
-  X  -  -  -  -  -  -
-  -  -  -  -  -  X  -
X  -  -  -  -  -  -  -
-  -  X  -  -  -  -  -
-  -  -  -  X  -  -  -
-  -  -  -  -  -  -  X
-  -  -  X  -  -  -  -  [4, 2, 7, 3, 6, 8, 1, 5]

-  -  -  -  -  -  X  -
-  X  -  -  -  -  -  -
-  -  -  X  -  -  -  -
X  -  -  -  -  -  -  -
-  -  -  -  -  -  -  X
-  -  -  -  X  -  -  -
-  -  X  -  -  -  -  -
-  -  -  -  -  X  -  -  [4, 2, 7, 3, 6, 8, 5, 1]

-  -  -  -  -  -  -  X
-  X  -  -  -  -  -  -
-  -  -  X  -  -  -  -
X  -  -  -  -  -  -  -
-  -  -  -  -  -  X  -
-  -  -  -  X  -  -  -
-  -  X  -  -  -  -  -
-  -  -  -  -  X  -  -  [4, 2, 7, 5, 1, 8, 6, 3]

-  -  -  -  X  -  -  -
-  X  -  -  -  -  -  -
-  -  -  -  -  -  -  X
X  -  -  -  -  -  -  -
-  -  -  X  -  -  -  -
-  -  -  -  -  -  X  -
-  -  X  -  -  -  -  -
-  -  -  -  -  X  -  -  [4, 2, 8, 5, 7, 1, 3, 6]

-  -  -  -  -  X  -  -
-  X  -  -  -  -  -  -
-  -  -  -  -  -  X  -
X  -  -  -  -  -  -  -
-  -  -  X  -  -  -  -
-  -  -  -  -  -  -  X
-  -  -  -  X  -  -  -
-  -  X  -  -  -  -  -  [4, 2, 8, 6, 1, 3, 5, 7]

-  -  -  -  X  -  -  -
-  X  -  -  -  -  -  -
```

```
-  -  -  -  -  X  -  -
X  -  -  -  -  -  -  -
-  -  -  -  -  -  X  -
-  -  -  X  -  -  -  -
-  -  -  -  -  -  -  X
-  -  X  -  -  -  -  - [4, 6, 1, 5, 2, 8, 3, 7]

-  -  X  -  -  -  -  -
-  -  -  -  X  -  -  -
-  -  -  -  -  -  X  -
X  -  -  -  -  -  -  -
-  -  -  X  -  -  -  -
-  X  -  -  -  -  -  -
-  -  -  -  -  -  -  X
-  -  -  -  -  X  -  - [4, 6, 8, 2, 7, 1, 3, 5]

-  -  -  -  -  X  -  -
-  -  -  X  -  -  -  -
-  -  -  -  -  -  X  -
X  -  -  -  -  -  -  -
-  -  -  -  -  -  -  X
-  X  -  -  -  -  -  -
-  -  -  -  X  -  -  -
-  -  X  -  -  -  -  - [4, 6, 8, 3, 1, 7, 5, 2]

-  -  -  -  X  -  -  -
-  -  -  -  -  -  -  X
-  -  -  X  -  -  -  -
X  -  -  -  -  -  -  -
-  -  -  -  -  -  X  -
-  X  -  -  -  -  -  -
-  -  -  -  -  X  -  -
-  -  X  -  -  -  -  - [4, 7, 1, 8, 5, 2, 6, 3]

-  -  X  -  -  -  -  -
-  -  -  -  -  X  -  -
-  -  -  -  -  -  -  X
X  -  -  -  -  -  -  -
-  -  -  -  X  -  -  -
-  -  -  -  -  -  X  -
-  X  -  -  -  -  -  -
-  -  -  X  -  -  -  - [4, 7, 3, 8, 2, 5, 1, 6]

-  -  -  -  -  -  X  -
-  -  -  -  X  -  -  -
-  -  X  -  -  -  -  -
X  -  -  -  -  -  -  -
-  -  -  -  -  X  -  -
```

```
-  -  -  -  -  -  -  X
-  X  -  -  -  -  -  -
-  -  -  X  -  -  -  - [4, 7, 5, 2, 6, 1, 3, 8]

-  -  -  -  -  X  -  -
-  -  -  X  -  -  -  -
-  -  -  -  -  -  X  -
X  -  -  -  -  -  -  -
-  -  X  -  -  -  -  -
-  -  -  -  X  -  -  -
-  X  -  -  -  -  -  -
-  -  -  -  -  -  -  X [4, 7, 5, 3, 1, 6, 8, 2]

-  -  -  -  X  -  -  -
-  -  -  -  -  -  -  X
-  -  -  X  -  -  -  -
X  -  -  -  -  -  -  -
-  -  X  -  -  -  -  -
-  -  -  -  -  X  -  -
-  X  -  -  -  -  -  -
-  -  -  -  -  -  X  - [4, 8, 1, 3, 6, 2, 7, 5]

-  -  X  -  -  -  -  -
-  -  -  -  -  X  -  -
-  -  -  X  -  -  -  -
X  -  -  -  -  -  -  -
-  -  -  -  -  -  -  X
-  -  -  -  X  -  -  -
-  -  -  -  -  -  X  -
-  X  -  -  -  -  -  - [4, 8, 1, 5, 7, 2, 6, 3]

-  -  X  -  -  -  -  -
-  -  -  -  -  X  -  -
-  -  -  -  -  -  -  X
X  -  -  -  -  -  -  -
-  -  -  X  -  -  -  -
-  -  -  -  -  -  X  -
-  -  -  -  X  -  -  -
-  X  -  -  -  -  -  - [4, 8, 5, 3, 1, 7, 2, 6]

-  -  -  -  X  -  -  -
-  -  -  -  -  -  X  -
-  -  -  X  -  -  -  -
X  -  -  -  -  -  -  -
-  -  X  -  -  -  -  -
-  -  -  -  -  -  -  X
-  -  -  -  -  X  -  -
-  X  -  -  -  -  -  - [5, 1, 4, 6, 8, 2, 7, 3]
```

```
-  X  -  -  -  -  -  -
-  -  -  -  -  X  -  -
-  -  -  -  -  -  -  X
-  -  X  -  -  -  -  -
X  -  -  -  -  -  -  -
-  -  -  X  -  -  -  -
-  -  -  -  -  -  X  -
-  -  -  -  X  -  -  - [5, 1, 8, 4, 2, 7, 3, 6]

-  X  -  -  -  -  -  -
-  -  -  -  X  -  -  -
-  -  -  -  -  -  X  -
-  -  -  X  -  -  -  -
X  -  -  -  -  -  -  -
-  -  -  -  -  -  -  X
-  -  -  -  -  X  -  -
-  -  X  -  -  -  -  - [5, 1, 8, 6, 3, 7, 2, 4]

-  X  -  -  -  -  -  -
-  -  -  -  -  -  X  -
-  -  -  -  X  -  -  -
-  -  -  -  -  -  -  X
X  -  -  -  -  -  -  -
-  -  -  X  -  -  -  -
-  -  -  -  -  X  -  -
-  -  X  -  -  -  -  - [5, 2, 4, 6, 8, 3, 1, 7]

-  -  -  -  -  -  X  -
-  X  -  -  -  -  -  -
-  -  -  -  -  X  -  -
-  -  X  -  -  -  -  -
X  -  -  -  -  -  -  -
-  -  -  X  -  -  -  -
-  -  -  -  -  -  -  X
-  -  -  -  X  -  -  - [5, 2, 4, 7, 3, 8, 6, 1]

-  -  -  -  -  -  -  X
-  X  -  -  -  -  -  -
-  -  -  -  X  -  -  -
-  -  X  -  -  -  -  -
X  -  -  -  -  -  -  -
-  -  -  -  -  -  X  -
-  -  -  X  -  -  -  -
-  -  -  -  -  X  -  - [5, 2, 6, 1, 7, 4, 8, 3]

-  -  -  X  -  -  -  -
-  X  -  -  -  -  -  -
```

```
-  -  -  -  -  -  -  X
-  -  -  -  -  X  -  -
X  -  -  -  -  -  -  -
-  -  X  -  -  -  -  -
-  -  -  -  X  -  -  -
-  -  -  -  -  -  X  - [5, 2, 8, 1, 4, 7, 3, 6]

-  -  -  X  -  -  -  -
-  X  -  -  -  -  -  -
-  -  -  -  -  -  X  -
-  -  -  -  X  -  -  -
X  -  -  -  -  -  -  -
-  -  -  -  -  -  -  X
-  -  -  -  -  X  -  -
-  -  X  -  -  -  -  - [5, 3, 1, 6, 8, 2, 4, 7]

-  -  X  -  -  -  -  -
-  -  -  -  -  X  -  -
-  X  -  -  -  -  -  -
-  -  -  -  -  -  X  -
X  -  -  -  -  -  -  -
-  -  -  X  -  -  -  -
-  -  -  -  -  -  -  X
-  -  -  -  X  -  -  - [5, 3, 1, 7, 2, 8, 6, 4]

-  -  X  -  -  -  -  -
-  -  -  -  X  -  -  -
-  X  -  -  -  -  -  -
-  -  -  -  -  -  -  X
X  -  -  -  -  -  -  -
-  -  -  -  -  -  X  -
-  -  -  X  -  -  -  -
-  -  -  -  -  X  -  - [5, 3, 8, 4, 7, 1, 6, 2]

-  -  -  -  -  X  -  -
-  -  -  -  -  -  -  X
-  X  -  -  -  -  -  -
-  -  -  X  -  -  -  -
X  -  -  -  -  -  -  -
-  -  -  -  -  -  X  -
-  -  -  -  X  -  -  -
-  -  X  -  -  -  -  - [5, 7, 1, 3, 8, 6, 4, 2]

-  -  X  -  -  -  -  -
-  -  -  -  -  -  -  X
-  -  -  X  -  -  -  -
-  -  -  -  -  -  X  -
X  -  -  -  -  -  -  -
```

```
-  -  -  -  -  X  -  -
-  X  -  -  -  -  -  -
-  -  -  -  X  -  -  - [5, 7, 1, 4, 2, 8, 6, 3]

-  -  X  -  -  -  -  -
-  -  -  -  X  -  -  -
-  -  -  -  -  -  -  X
-  -  -  X  -  -  -  -
X  -  -  -  -  -  -  -
-  -  -  -  -  -  X  -
-  X  -  -  -  -  -  -
-  -  -  -  -  X  -  - [5, 7, 2, 4, 8, 1, 3, 6]

-  -  -  -  -  X  -  -
-  -  X  -  -  -  -  -
-  -  -  -  -  -  X  -
-  -  -  X  -  -  -  -
X  -  -  -  -  -  -  -
-  -  -  -  -  -  -  X
-  X  -  -  -  -  -  -
-  -  -  -  X  -  -  - [5, 7, 2, 6, 3, 1, 4, 8]

-  -  -  -  -  X  -  -
-  -  X  -  -  -  -  -
-  -  -  -  X  -  -  -
-  -  -  -  -  -  X  -
X  -  -  -  -  -  -  -
-  -  -  X  -  -  -  -
-  X  -  -  -  -  -  -
-  -  -  -  -  -  -  X [5, 7, 2, 6, 3, 1, 8, 4]

-  -  -  -  -  X  -  -
-  -  X  -  -  -  -  -
-  -  -  -  X  -  -  -
-  -  -  -  -  -  -  X
X  -  -  -  -  -  -  -
-  -  -  X  -  -  -  -
-  X  -  -  -  -  -  -
-  -  -  -  -  -  X  - [5, 7, 4, 1, 3, 8, 6, 2]

-  -  -  X  -  -  -  -
-  -  -  -  -  -  -  X
-  -  -  -  X  -  -  -
-  -  X  -  -  -  -  -
X  -  -  -  -  -  -  -
-  -  -  -  -  -  X  -
-  X  -  -  -  -  -  -
-  -  -  -  -  X  -  - [5, 8, 4, 1, 3, 6, 2, 7]
```

```
-  -  -  X  -  -  -  -
-  -  -  -  -  -  X  -
-  -  -  -  X  -  -  -
-  -  X  -  -  -  -  -
X  -  -  -  -  -  -  -
-  -  -  -  -  X  -  -
-  -  -  -  -  -  -  X
-  X  -  -  -  -  -  - [5, 8, 4, 1, 7, 2, 6, 3]

-  -  -  X  -  -  -  -
-  -  -  -  -  X  -  -
-  -  -  -  -  -  -  X
-  -  X  -  -  -  -  -
X  -  -  -  -  -  -  -
-  -  -  -  -  -  X  -
-  -  -  -  X  -  -  -
-  X  -  -  -  -  -  - [6, 1, 5, 2, 8, 3, 7, 4]

-  X  -  -  -  -  -  -
-  -  -  X  -  -  -  -
-  -  -  -  -  X  -  -
-  -  -  -  -  -  -  X
-  -  X  -  -  -  -  -
X  -  -  -  -  -  -  -
-  -  -  -  -  -  X  -
-  -  -  -  X  -  -  - [6, 2, 7, 1, 3, 5, 8, 4]

-  -  -  X  -  -  -  -
-  X  -  -  -  -  -  -
-  -  -  -  X  -  -  -
-  -  -  -  -  -  -  X
-  -  -  -  -  X  -  -
X  -  -  -  -  -  -  -
-  -  X  -  -  -  -  -
-  -  -  -  -  -  X  - [6, 2, 7, 1, 4, 8, 5, 3]

-  -  -  X  -  -  -  -
-  X  -  -  -  -  -  -
-  -  -  -  -  -  -  X
-  -  -  -  X  -  -  -
-  -  -  -  -  -  X  -
X  -  -  -  -  -  -  -
-  -  X  -  -  -  -  -
-  -  -  -  -  X  -  - [6, 3, 1, 7, 5, 8, 2, 4]

-  -  X  -  -  -  -  -
-  -  -  -  -  -  X  -
```

```
-  X  -  -  -  -  -  -
-  -  -  -  -  -  -  X
-  -  -  -  X  -  -  -
X  -  -  -  -  -  -  -
-  -  -  X  -  -  -  -
-  -  -  -  -  X  -  - [6, 3, 1, 8, 4, 2, 7, 5]

-  -  X  -  -  -  -  -
-  -  -  -  -  X  -  -
-  X  -  -  -  -  -  -
-  -  -  -  X  -  -  -
-  -  -  -  -  -  -  X
X  -  -  -  -  -  -  -
-  -  -  -  -  -  X  -
-  -  -  X  -  -  -  - [6, 3, 1, 8, 5, 2, 4, 7]

-  -  X  -  -  -  -  -
-  -  -  -  -  X  -  -
-  X  -  -  -  -  -  -
-  -  -  -  -  -  X  -
-  -  -  -  X  -  -  -
X  -  -  -  -  -  -  -
-  -  -  -  -  -  -  X
-  -  -  X  -  -  -  - [6, 3, 5, 7, 1, 4, 2, 8]

-  -  -  -  X  -  -  -
-  -  -  -  -  -  X  -
-  X  -  -  -  -  -  -
-  -  -  -  -  X  -  -
-  -  X  -  -  -  -  -
X  -  -  -  -  -  -  -
-  -  -  X  -  -  -  -
-  -  -  -  -  -  -  X [6, 3, 5, 8, 1, 4, 2, 7]

-  -  -  -  X  -  -  -
-  -  -  -  -  -  X  -
-  X  -  -  -  -  -  -
-  -  -  -  -  X  -  -
-  -  X  -  -  -  -  -
X  -  -  -  -  -  -  -
-  -  -  -  -  -  -  X
-  -  -  X  -  -  -  - [6, 3, 7, 2, 4, 8, 1, 5]

-  -  -  -  -  -  X  -
-  -  -  X  -  -  -  -
-  X  -  -  -  -  -  -
-  -  -  -  X  -  -  -
-  -  -  -  -  -  -  X
```

```
X  -  -  -  -  -  -  -
-  -  X  -  -  -  -  -
-  -  -  -  -  X  -  - [6, 3, 7, 2, 8, 5, 1, 4]

-  -  -  -  -  -  X  -
-  -  -  X  -  -  -  -
-  X  -  -  -  -  -  -
-  -  -  -  -  -  -  X
-  -  -  -  -  X  -  -
X  -  -  -  -  -  -  -
-  -  X  -  -  -  -  -
-  -  -  -  X  -  -  - [6, 3, 7, 4, 1, 8, 2, 5]

-  -  -  -  X  -  -  -
-  -  -  -  -  -  X  -
-  X  -  -  -  -  -  -
-  -  -  X  -  -  -  -
-  -  -  -  -  -  -  X
X  -  -  -  -  -  -  -
-  -  X  -  -  -  -  -
-  -  -  -  -  X  -  - [6, 4, 1, 5, 8, 2, 7, 3]

-  -  X  -  -  -  -  -
-  -  -  -  -  X  -  -
-  -  -  -  -  -  -  X
-  X  -  -  -  -  -  -
-  -  -  X  -  -  -  -
X  -  -  -  -  -  -  -
-  -  -  -  -  -  X  -
-  -  -  -  X  -  -  - [6, 4, 2, 8, 5, 7, 1, 3]

-  -  -  -  -  -  X  -
-  -  X  -  -  -  -  -
-  -  -  -  -  -  -  X
-  X  -  -  -  -  -  -
-  -  -  -  X  -  -  -
X  -  -  -  -  -  -  -
-  -  -  -  -  X  -  -
-  -  -  X  -  -  -  - [6, 4, 7, 1, 3, 5, 2, 8]

-  -  -  X  -  -  -  -
-  -  -  -  -  -  X  -
-  -  -  -  X  -  -  -
-  X  -  -  -  -  -  -
-  -  -  -  -  X  -  -
X  -  -  -  -  -  -  -
-  -  X  -  -  -  -  -
-  -  -  -  -  -  -  X [6, 4, 7, 1, 8, 2, 5, 3]
```

```
-  -  -  X  -  -  -  -
-  -  -  -  -  X  -  -
-  -  -  -  -  -  -  X
-  X  -  -  -  -  -  -
-  -  -  -  -  -  X  -
X  -  -  -  -  -  -  -
-  -  X  -  -  -  -  -
-  -  -  -  X  -  -  - [6, 8, 2, 4, 1, 7, 5, 3]

-  -  -  -  X  -  -  -
-  -  X  -  -  -  -  -
-  -  -  -  -  -  -  X
-  -  -  X  -  -  -  -
-  -  -  -  -  -  X  -
X  -  -  -  -  -  -  -
-  -  -  -  -  X  -  -
-  X  -  -  -  -  -  - [7, 1, 3, 8, 6, 4, 2, 5]

-  X  -  -  -  -  -  -
-  -  -  -  -  -  X  -
-  -  X  -  -  -  -  -
-  -  -  -  -  X  -  -
-  -  -  -  -  -  -  X
-  -  -  -  X  -  -  -
X  -  -  -  -  -  -  -
-  -  -  X  -  -  -  - [7, 2, 4, 1, 8, 5, 3, 6]

-  -  -  X  -  -  -  -
-  X  -  -  -  -  -  -
-  -  -  -  -  -  X  -
-  -  X  -  -  -  -  -
-  -  -  -  -  X  -  -
-  -  -  -  -  -  -  X
X  -  -  -  -  -  -  -
-  -  -  -  X  -  -  - [7, 2, 6, 3, 1, 4, 8, 5]

-  -  -  -  X  -  -  -
-  X  -  -  -  -  -  -
-  -  -  X  -  -  -  -
-  -  -  -  -  X  -  -
-  -  -  -  -  -  -  X
-  -  X  -  -  -  -  -
X  -  -  -  -  -  -  -
-  -  -  -  -  -  X  - [7, 3, 1, 6, 8, 5, 2, 4]

-  -  X  -  -  -  -  -
-  -  -  -  -  -  X  -
```

```
-  X  -  -  -  -  -  -
-  -  -  -  -  -  -  X
-  -  -  -  -  X  -  -
-  -  -  X  -  -  -  -
X  -  -  -  -  -  -  -
-  -  -  -  X  -  -  - [7, 3, 8, 2, 5, 1, 6, 4]

-  -  -  -  -  X  -  -
-  -  -  X  -  -  -  -
-  X  -  -  -  -  -  -
-  -  -  -  -  -  -  X
-  -  -  -  X  -  -  -
-  -  -  -  -  -  X  -
X  -  -  -  -  -  -  -
-  -  X  -  -  -  -  - [7, 4, 2, 5, 8, 1, 3, 6]

-  -  -  -  -  X  -  -
-  -  X  -  -  -  -  -
-  -  -  -  -  -  X  -
-  X  -  -  -  -  -  -
-  -  -  X  -  -  -  -
-  -  -  -  -  -  -  X
X  -  -  -  -  -  -  -
-  -  -  -  X  -  -  - [7, 4, 2, 8, 6, 1, 3, 5]

-  -  -  -  -  X  -  -
-  -  X  -  -  -  -  -
-  -  -  -  -  -  X  -
-  X  -  -  -  -  -  -
-  -  -  -  -  -  -  X
-  -  -  -  X  -  -  -
X  -  -  -  -  -  -  -
-  -  -  X  -  -  -  - [7, 5, 3, 1, 6, 8, 2, 4]

-  -  -  X  -  -  -  -
-  -  -  -  -  -  X  -
-  -  X  -  -  -  -  -
-  -  -  -  -  -  -  X
-  X  -  -  -  -  -  -
-  -  -  -  X  -  -  -
X  -  -  -  -  -  -  -
-  -  -  -  -  X  -  - [8, 2, 4, 1, 7, 5, 3, 6]

-  -  -  X  -  -  -  -
-  X  -  -  -  -  -  -
-  -  -  -  -  -  X  -
-  -  X  -  -  -  -  -
-  -  -  -  -  X  -  -
```

```
-  -  -  -  -  -  -  X
-  -  -  -  X  -  -  -
X  -  -  -  -  -  -  - [8, 2, 5, 3, 1, 7, 4, 6]

-  -  -  -  X  -  -  -
-  X  -  -  -  -  -  -
-  -  -  X  -  -  -  -
-  -  -  -  -  -  -  X  -
-  -  X  -  -  -  -  -
-  -  -  -  -  -  -  X
-  -  -  -  -  -  X  -
X  -  -  -  -  -  -  - [8, 3, 1, 6, 2, 5, 7, 4]

-  -  X  -  -  -  -  -
-  -  -  -  X  -  -  -
-  X  -  -  -  -  -  -
-  -  -  -  -  -  -  X
-  -  -  -  -  -  X  -
-  -  -  X  -  -  -  -
-  -  -  -  -  -  -  X  -
X  -  -  -  -  -  -  - [8, 4, 1, 3, 6, 2, 7, 5]

-  -  X  -  -  -  -  -
-  -  -  -  -  -  X  -
-  -  -  X  -  -  -  -
-  X  -  -  -  -  -  -
-  -  -  -  -  -  -  X
-  -  -  -  X  -  -  -
-  -  -  -  -  -  -  X  -
X  -  -  -  -  -  -  -
```

[13]:
```
escribe_solucion([6, 4, 1, 5, 8, 2, 7, 3])
```

```
-  -  X  -  -  -  -  -
-  -  -  -  -  X  -  -
-  -  -  -  -  -  -  X
-  X  -  -  -  -  -  -
-  -  -  X  -  -  -  -
X  -  -  -  -  -  -  -
-  -  -  -  -  -  X  -
-  -  -  -  X  -  -  -
```

[ ]:
```

```

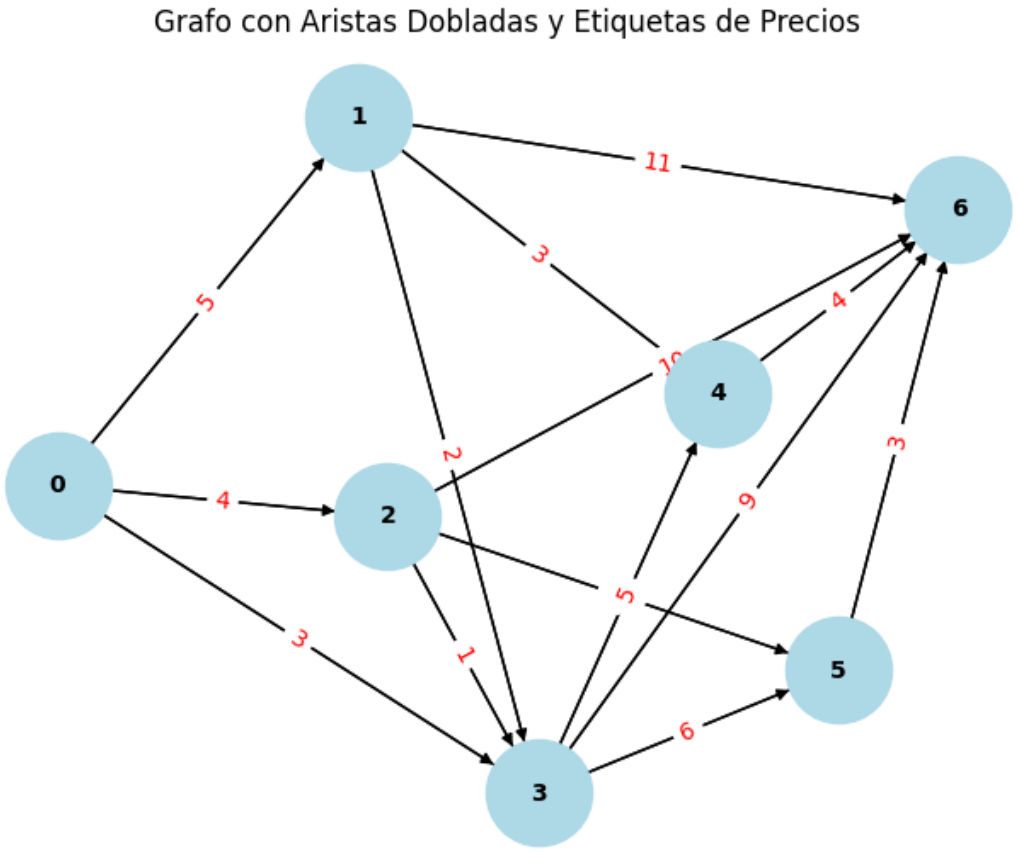
[27]:
```
#Viaje por el rio - Programación dinámica
##########################################################################################

```

```
TARIFAS = [
[0,5,4,3,999,999,999],
[999,0,999,2,3,999,11],
[999,999, 0,1,999,4,10],
[999,999,999, 0,5,6,9],
[999,999, 999,999,0,999,4],
[999,999, 999,999,999,0,3],
[999,999,999,999,999,999,0]
]

#999 se puede sustituir por float("inf")


#Calculo de la matriz de PRECIOS y RUTAS
#####################################################################
def Precios(TARIFAS):
#####################################################################
  #Total de Nodos
  N = len(TARIFAS[0])

  #Inicialización de la tabla de precios
  PRECIOS = [ [9999]*N for i in [9999]*N]
  RUTA = [ [""]*N for i in [""]*N]

  for i in range(0,N-1):
    RUTA[i][i] = i              #Para ir de i a i se "pasa por i"
    PRECIOS[i][i] = 0           #Para ir de i a i se se paga 0
    for j in range(i+1, N):
      MIN = TARIFAS[i][j]
      RUTA[i][j] = i

      for k in range(i, j):
        if PRECIOS[i][k] + TARIFAS[k][j] < MIN:
            MIN = min(MIN, PRECIOS[i][k] + TARIFAS[k][j] )
            RUTA[i][j] = k          #Anota que para ir de i a j hay que pasar␣
 ↪por k
       PRECIOS[i][j] = MIN

  return PRECIOS,RUTA
#####################################################################

PRECIOS,RUTA = Precios(TARIFAS)
#print(PRECIOS[0][6])
```

[47]:
```
import networkx as nx
import matplotlib.pyplot as plt
```

```
# TARIFAS matriz (Matriz de adyacencia)
TARIFAS = [
[0,5,4,3,999,999,999],
[999,0,999,2,3,999,11],
[999,999, 0,1,999,4,10],
[999,999,999, 0,5,6,9],
[999,999, 999,999,0,999,4],
[999,999, 999,999,999,0,3],
[999,999,999,999,999,999,0]
]

# Función para graficar el grafo con aristas dobladas
def plot_graph(tarifas):
    G = nx.DiGraph()  # Crear un grafo dirigido

    # Agregar aristas con pesos (precios) desde la matriz de tarifas
    num_nodes = len(tarifas)  # Número de nodos
    for i in range(num_nodes):
        for j in range(num_nodes):
            if tarifas[i][j] != 999 and i != j:  # Solo agregar conexiones␣
 ↪válidas
                G.add_edge(i, j, weight=tarifas[i][j])  # Nodos numerados desde␣
 ↪0

    # Definir posiciones manualmente de los nodos (según la disposición␣
 ↪solicitada)
    pos = {
        0: (0, 0),        # Nodo 0 en la columna más a la izquierda
        1: (1, 1.2),        # Nodo 1 en la segunda columna, fila 1
        2: (1.1, -0.1),        # Nodo 2 en la segunda columna, fila 0
        3: (1.6, -1),       # Nodo 3 en la segunda columna, fila -1
        4: (2.2, 0.3),      # Nodo 4 en la tercera columna, entre filas
        5: (2.6, -0.6),     # Nodo 5 en la tercera columna, entre filas
        6: (3, 0.9),        # Nodo 6 en la columna más a la derecha
    }

    # Dibujar los nodos
    nx.draw(G, pos, with_labels=True, node_color='lightblue', node_size=2000,␣
 ↪font_size=10, font_weight='bold')

    # Obtener las aristas y sus pesos (precios)
    edge_labels = nx.get_edge_attributes(G, 'weight')

    # Dibujar aristas con curvatura
    curved_edges = [(u, v) for u, v in G.edges() if G.has_edge(v, u)]  #␣
 ↪Aristas que deben estar dobladas
```

```
    straight_edges = list(set(G.edges()) - set(curved_edges))  # Aristas que␣
↪pueden ser rectas

    # Dibujar las aristas rectas
    nx.draw_networkx_edges(G, pos, edgelist=straight_edges, arrowstyle='-|>',␣
↪arrowsize=20, edge_color='black')

    # Dibujar las aristas curvas
    arc_rad = 0.25  # Radio de curvatura para las aristas
    nx.draw_networkx_edges(G, pos, edgelist=curved_edges,␣
↪connectionstyle=f'arc3,rad={arc_rad}', arrowstyle='-|>', arrowsize=20,␣
↪edge_color='black')

    # Dibujar etiquetas en las aristas (los precios)
    nx.draw_networkx_edge_labels(G, pos, edge_labels=edge_labels,␣
↪font_color='red')

    # Mostrar el gráfico
    plt.title('Grafo con Aristas Dobladas y Etiquetas de Precios')
    plt.show()

# Llamar a la función para graficar el grafo
plot_graph(TARIFAS)
```



[49]:
```
print("PRECIOS")
for i in range(len(TARIFAS)):
  print(PRECIOS[i])

print("\nRUTA")
for i in range(len(TARIFAS)):
  print(RUTA[i])

#Determinar la ruta con Recursividad
def calcular_ruta(RUTA, desde, hasta):
  if desde == hasta:
    #print("Ir a :" + str(desde))
    return ""
  else:
    return str(calcular_ruta( RUTA, desde, RUTA[desde][hasta])) +  \
                ',' + \
                str(RUTA[desde][hasta] \
              )
```

```
print("\nLa ruta es:")
calcular_ruta(RUTA, 0, 6)
```

```
PRECIOS
[0, 5, 4, 3, 8, 8, 11]
[9999, 0, 999, 2, 3, 8, 7]
[9999, 9999, 0, 1, 6, 4, 7]
[9999, 9999, 9999, 0, 5, 6, 9]
[9999, 9999, 9999, 9999, 0, 999, 4]
[9999, 9999, 9999, 9999, 9999, 0, 3]
[9999, 9999, 9999, 9999, 9999, 9999, 9999]

RUTA
[0, 0, 0, 0, 1, 2, 5]
['', 1, 1, 1, 1, 3, 4]
['', '', 2, 2, 3, 2, 5]
['', '', '', 3, 3, 3, 3]
['', '', '', '', 4, 4, 4]
['', '', '', '', '', 5, 5]
['', '', '', '', '', '', '']

La ruta es:
```

[49]: ',0,2,5'

[ ]: