Análisis y Diseño de Algoritmos

Sistema de Asignación de Mecánicos y Averías

Francisco Javier Mercader Martínez Pedro Alarcón Fuentes

En esta memoria se explicará el código utilizado para crear una sistema para la asignación de mecánicos a diferentes averías, evaluando las capacidades de los mecánicos para reparar averías específicas y optimizando el proceso de asignación. La estructura permite procesar varios casos de prueba y validar los resultados de cada asignación.

Índice de Funciones

---

1. Función `encontrar_archivos_in`

```python
def encontrar_archivos_in(directorio='.'):
    """
    Busca todos los archivos .in en el directorio especificado.

    :param directorio: Ruta del directorio donde buscar archivos `.in`.
    :return: Lista de rutas de archivos `.in` encontradas.
    """
    archivos_in = []
    for root, _, files in os.walk(directorio):
        for file in files:
            if file.endswith('.in'):
                archivos_in.append(os.path.join(root, file))
    return archivos_in
```

Descripción: Busca y devuelve una lista de archivos con extensión **`.in`** en el directorio especificado, que se utilizarán como entradas de datos para el programa.

---

2. Función `encontrar_archivos_out`

```python
def encontrar_archivos_out(directorio='.'):
    """
    Busca todos los archivos .out en el directorio especificado.

    :param directorio: Ruta del directorio donde buscar archivos `.out`.
    :return: Lista de rutas de archivos `.out` encontradas.
    """
    archivos_out = []
    for root, _, files in os.walk(directorio):
        for file in files:
            if file.endswith('.out'):
                archivos_out.append(os.path.join(root, file))
    return archivos_out
```

Descripción: Busca y devuelve una lista de archivos con extensión **`.out`** en el directorio especificado, que se utilizarán para realizar la validación de los resultados obtenidos.

---

3. Función `leer_entrada`

```python
def leer_entrada(file_path):
    """
    Lee el archivo de entrada y convierte la información en una estructura de
    datos adecuada.

    :param file_path: Ruta del archivo de entrada.
    :return: Número de casos de prueba (P) y lista de casos de prueba.
    """
    with open(file_path, 'r') as file:
        lineas = file.readlines()

    P = int(lineas[0])
    casos = []

    indice = 1
    for _ in range(P):
        M, A = map(int, lineas[indice].split())
        indice += 1

        capacidades = []
        for i in range(M):
            capacidades.append(list(map(int, lineas[indice].split())))
            indice += 1

        casos.append({'M': M, 'A': A, 'capacidades': capacidades})

    return P, casos
```

Descripción y objetivo de la función: La función **leer_entrada** se encarga de leer un archivo de entrada **.in** y extraer los datos necesarios para cada caso de prueba. Almacena el número de mecánicos y averías, así como las capacidades de cada mecánico para reparar averías específicas en una estructura de datos organizada.

Parámetros:

- **file_path** : La ruta al archivo de entrada

Proceso:

1. Lee todas las líneas del archivo de entrada y extrae el número de casos de prueba.
2. Para cada caso, lee el número de mecánicos (**M**) y averías (**A**).
3. Crea una matriz **capacidades** que indica las habilidades de cada mecánico para reparar las averías.
4. Almacena cada caso en una lista de diccionarios.

---

4. Función voraz

```
def voraz(c, num_averias, mecanicos):
    '''
    Función que usa el algoritmo voraz visto en clase adaptado a este problema
    '''
    s = []
    while len(c) > 0 and not solution(s, num_averias):
        x = select(c, mecanicos)
        c.remove(x)
        if factible(s, x):
            s.append(x)
    return s
```

Descripción: Esta función implementa un algoritmo voraz para seleccionar las asignaciones de mecánicos y averías. La función itera sobre una lista de candidatos (pares de mecánico y avería) y selecciona aquellos que son factibles hasta que se encuentra una solución o se agotan los candidatos.

Parámetros:

- **c**: Lista de candidatos (pares de mecánico y avería).
- **num_averias**: Número total de averías.
- **mecanicos**: Matriz de capacidades de los mecánicos.

Proceso:

1. Inicializa una lista vacía **s** para almacenar las asignaciones.
2. Mientras haya candidatos y no se haya encontrado una solución:
    - Selecciona un candidato **x** usando la función **select**.
    - Elimina el candidato seleccionado de la lista **c**.
    - Si el candidato es factible (usando la función **factible**), se agrega a la lista de asignaciones **s**.

---

5. Función `factible`

```
def factible(mecanico, averia, capacidades, asignaciones):
    """
    Verifica si un mecánico puede ser asignado a una avería.

    mecanico: Índice del mecánico.
    averia: Índice de la avería.
    capacidades: Matriz C que indica las capacidades de los mecánicos.
    asignaciones: Lista que indica si una avería ya fue asignada.

    Retorna True si el mecánico puede reparar la avería y aún no ha sido asignada.
    """
    return capacidades[mecanico][averia] == 1 and asignaciones[averia] == 0
```

Descripción: Verifica si un mecánico específico puede ser asignado a una avería dada. Esto depende de la capacidad del mecánico para repararla y de si la avería ha sido asignada.

Parámetros:

- **`mecanico`**: Índice del mecánico.
- **`averia`**: Índice de la avería.
- **`capacidades`**: Matriz que representa las capacidades de los mecánicos.
- **`asignaciones`**: Lista de asignaciones para controlar qué averías han sido ya asignadas.

---

6. Función `select`

```
def select(mecanico, capacidades, asignaciones, A):
    """
    Selecciona la mejor avería que un mecánico puede reparar, si es posible.

    mecanico: Índice del mecánico.
    capacidades: Matriz C que indica las capacidades de los mecánicos.
    asignaciones: Lista que indica si una avería ya fue asignada.
    A: Número de averías.

    Retorna el índice de la avería seleccionada o -1 si no hay ninguna disponible.
    """
    for averia in range(A):
        if factible(mecanico, averia, capacidades, asignaciones):
            return averia
    return -1
```

Descripción: Esta función busca una avería que el mecánico puede reparar. Si encuentra una, devuelve el índice de la avería; si no, devuelve **`-1`**.

Proceso:

1. Recorre cada avería posible.
2. Usa la función **`factible`** para verificar si el mecánico puede asignarse a ella.
3. Devuelve el índice de la primera avería factible.

---

7. Función `solution`

```python
def solution(P, casos):
    """
    Resuelve el problema para P casos de prueba.

    P: Número de casos de prueba.
    casos: Lista de casos de prueba, cada uno con M mecánicos, A averías y la matriz de

    Retorna una lista con las soluciones de cada caso.
    """
    resultados = []

    for caso in casos:
        M, A, capacidades = caso['M'], caso['A'], caso['capacidades']

        asignaciones = [0] * A  # Lista para rastrear qué mecánico se asigna a cada averí
        averias_reparadas = 0

        for mecanico in range(M):
            averia_seleccionada = select(mecanico, capacidades, asignaciones, A)
            if averia_seleccionada != -1:
                asignaciones[averia_seleccionada] = mecanico + 1  # Asigna el mecánico (i
                averias_reparadas += 1

        # Guardamos el resultado del caso
        resultados.append((averias_reparadas, asignaciones))

    return resultados
```

Descripción y objetivo de la función: **`solution`** implementa el proceso de asignación para cada caso de prueba. Para cada mecánico, selecciona una avería que pueda reparar, y si la encuentra, se asigna y actualiza el número total de averías reparadas.

Proceso:

1. Para cada caso, inicializa **`asignaciones`** con `0` para indicar avería ha sido asignada.
2. Para cada mecánico, selecciona la avería adecuada mediante la función **`select`**.
3. Lleva un conteo de averías asignadas y almacena el resultado de cada caso en una lista.

---

8. Función procesar_matriz

```
def procesar_matriz(matriz):
    '''
    Esta función procesa la matriz dada y saca el número de averias, candidatos y usa el
    voraz para obtener una solución
    '''
    num_averias = matriz.shape[1]
    candidatos = [(i, j) for i in range(matriz.shape[0]) for j in range(matriz.shape[1]) i
    solucion = voraz(candidatos, num_averias, matriz)
    resultado = [0] * num_averias
    for i, j in solucion:
        resultado[j] = i + 1 # Asignar el mecánico (i+1) a la avería (j)
    return resultado
```

Descripción y objetivo de la función: **procesar_matriz** toma una matriz de capacidades y aplica el algoritmo voraz para generar una lista de asignaciones de mecánicos

---

9. Función generar_salida

```
def generar_salida(matrices):
    casos = []
    for matriz in matrices:
        M = len(matriz)
        A = len(matriz[0]) if M > 0 else 0
        capacidades = matriz

        casos.append({'M': M, 'A': A, 'capacidades': capacidades})
    resultados = solution([], casos)
    return resultados
```

Descripción: **generar_salida** organiza los resultados para múltiples matrices de capacidades, aplicando **solution**.

Resultado

```
file_paths_in = encontrar_archivos_in('.')
file_paths_out = encontrar_archivos_out('.')

P, casos = leer_entrada(file_paths_in[0])
matrices = [caso['capacidades'] for caso in casos]
resultados = generar_salida(matrices)
print(P)
```

```
## 20
```

```
for resultado in resultados:
  print(resultado[0])
  print(textwrap.fill(' '.join(map(str, resultado[1])), width=86))
```

```
## 4
## 1 2 3 5
## 7
## 1 2 3 5 4 9 7 0
## 3
## 1 2 3 0 0
## 7
## 1 4 2 5 3 7 6 0 0 0 0
## 1
## 0 2
## 5
## 1 3 2 6 4 0
## 4
## 3 2 4 1 0 0 0 0 0 0 0 0 0 0 0
## 2
## 1 6
## 11
## 1 2 6 4 3 8 9 7 5 0 10 11 0 0 0 0
## 8
## 3 2 4 1 5 6 7 8
## 4
## 0 2 1 4 3 0 0 0 0 0 0 0 0 0 0
## 2
## 1 0 2
## 9
## 3 1 6 2 4 5 8 7 9
## 3
## 1 2 4
## 11
## 2 3 1 4 7 5 8 10 9 11 6 0
## 6
## 3 1 2 4 5 6
## 10
## 2 1 6 3 7 5 8 10 4 11
## 2
## 1 4 0
## 8
## 1 2 3 4 5 8 6 7 0 0 0 0 0 0 0 0
## 11
## 1 2 4 6 3 5 9 7 13 8 10
```

```
print()
```

Análisis de los resultados con el validador

```
for file_path in file_paths_in:
  output_name = os.path.splitext(file_path)[0] + '_out.txt'
  output_file = os.path.join(directorio, output_name)

  P, casos = leer_entrada(file_path)

  matrices = [caso['capacidades'] for caso in casos]
  resultados = generar_salida(matrices)

  with open(output_file, 'w') as file:
    file.write(f"{P}\n")
    for resultado in resultados:
      file.write(f"{resultado[0]}\n")
      file.write(" ".join(map(str, resultado[1])) + "\n")

  corresponding_out_file = os.path.splitext(file_path)[0] + '.out'
  if os.path.exists(corresponding_out_file):
    print(f"\nValidando {corresponding_out_file}...")
    validador_E_AR(fichero_entrada=file_path,
                   fichero_salida=output_file,
                   fichero_salida_profesor=corresponding_out_file)
  else:
    print(f"Archivo de salida del profesor no encontrado para {file_path}")
```

```
## Validando 701a.out...
## En el caso 0: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 1: Todo correcto y el porcentaje de averías reparadas (7,8) es del 85.71%.
## En el caso 2: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 3: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 4: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 5: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 6: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 7: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 8: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 9: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 10: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 11: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 12: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 13: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 14: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 15: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 16: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 17: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 18: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 19: Todo correcto y el porcentaje de averías reparadas es del 100%.
## La media porcentual para 20 casos es del 99.29%.

## Validando 701b.out...
## En el caso 0: Todo correcto y el porcentaje de averías reparadas es del 100%.
```

```
## En el caso 1: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 2: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 3: Todo correcto y el porcentaje de averías reparadas (7,9) es del 71.43%.
## En el caso 4: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 5: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 6: Todo correcto y el porcentaje de averías reparadas (9,10) es del 88.89%.
## En el caso 7: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 8: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 9: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 10: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 11: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 12: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 13: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 14: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 15: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 16: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 17: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 18: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 19: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 20: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 21: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 22: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 23: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 24: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 25: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 26: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 27: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 28: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 29: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 30: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 31: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 32: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 33: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 34: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 35: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 36: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 37: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 38: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 39: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 40: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 41: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 42: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 43: Todo correcto y el porcentaje de averías reparadas (4,5) es del 75.0%.
## En el caso 44: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 45: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 46: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 47: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 48: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 49: Todo correcto y el porcentaje de averías reparadas es del 100%.
## La media porcentual para 50 casos es del 98.71%.

## Validando 701c.out...
```

```
## En el caso 0: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 1: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 2: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 3: Todo correcto y el porcentaje de averías reparadas (26,27) es del 96.15%.
## En el caso 4: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 5: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 6: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 7: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 8: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 9: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 10: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 11: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 12: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 13: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 14: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 15: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 16: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 17: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 18: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 19: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 20: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 21: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 22: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 23: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 24: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 25: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 26: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 27: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 28: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 29: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 30: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 31: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 32: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 33: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 34: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 35: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 36: Todo correcto y el porcentaje de averías reparadas (11,12) es del 90.91%
## En el caso 37: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 38: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 39: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 40: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 41: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 42: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 43: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 44: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 45: Todo correcto y el porcentaje de averías reparadas (18,19) es del 94.44%
## En el caso 46: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 47: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 48: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 49: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 50: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 51: Todo correcto y el porcentaje de averías reparadas es del 100%.
```

```
## En el caso 52: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 53: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 54: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 55: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 56: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 57: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 58: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 59: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 60: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 61: Todo correcto y el porcentaje de averías reparadas (19,20) es del 94.74%
## En el caso 62: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 63: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 64: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 65: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 66: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 67: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 68: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 69: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 70: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 71: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 72: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 73: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 74: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 75: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 76: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 77: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 78: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 79: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 80: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 81: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 82: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 83: Todo correcto y el porcentaje de averías reparadas (16,17) es del 93.75%
## En el caso 84: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 85: Todo correcto y el porcentaje de averías reparadas (9,10) es del 88.89%.
## En el caso 86: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 87: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 88: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 89: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 90: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 91: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 92: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 93: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 94: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 95: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 96: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 97: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 98: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 99: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 100: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 101: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 102: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 103: Todo correcto y el porcentaje de averías reparadas es del 100%.
```

```
## En el caso 104: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 105: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 106: Todo correcto y el porcentaje de averías reparadas (26,27) es del 96.15
## En el caso 107: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 108: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 109: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 110: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 111: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 112: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 113: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 114: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 115: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 116: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 117: Todo correcto y el porcentaje de averías reparadas (32,33) es del 96.88
## En el caso 118: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 119: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 120: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 121: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 122: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 123: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 124: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 125: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 126: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 127: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 128: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 129: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 130: Todo correcto y el porcentaje de averías reparadas (38,39) es del 97.37
## En el caso 131: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 132: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 133: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 134: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 135: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 136: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 137: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 138: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 139: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 140: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 141: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 142: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 143: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 144: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 145: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 146: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 147: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 148: Todo correcto y el porcentaje de averías reparadas es del 100%.
## En el caso 149: Todo correcto y el porcentaje de averías reparadas es del 100%.
## La media porcentual para 150 casos es del 99.66%.
```

```python
def analyze_complexity(file_paths):
    """
    Mide el tiempo de ejecución para cada archivo y genera un gráfico comparativo.
```

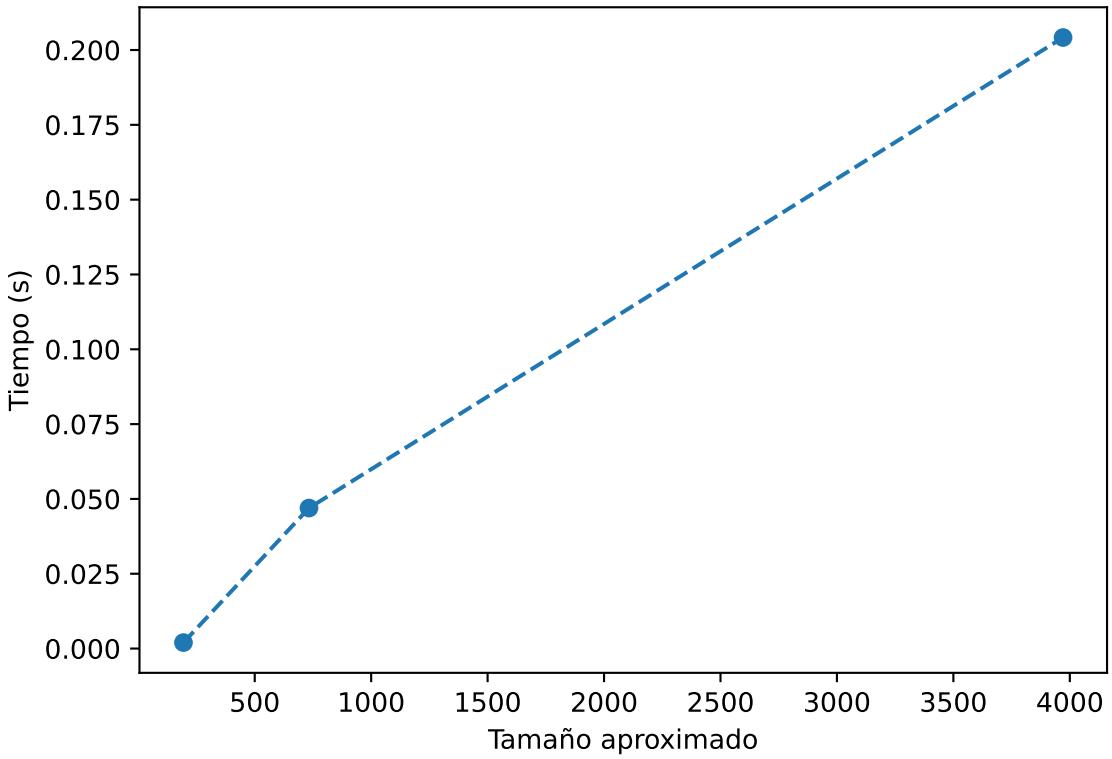
```
    :param file_paths: Lista de rutas de archivos `.in`.
    """
    sizes = []
    times = []

    for file_path in file_paths:
        # Medir el tiempo de ejecución para leer y procesar cada archivo
        start_time = time.time()

        # Leer la entrada y procesar los casos de prueba
        P, casos = leer_entrada(file_path)
        resultados = solution(P, casos)

        end_time = time.time()
        elapsed_time = end_time - start_time

        # Calcular el tamaño aproximado de la entrada
        with open(file_path, 'r') as f:
            size = sum(1 for _ in f)

        # Agregar el tamaño y tiempo a las listas
        sizes.append(size)
        times.append(elapsed_time)

    # Graficar el resultado comparativo para todos los archivos
    plt.plot(sizes, times, marker='o', linestyle='--')
    plt.xlabel('Tamaño aproximado')
    plt.ylabel('Tiempo (s)')
    plt.show()
```

Esta función mide el tiempo de ejecución para cada archivo en **file_paths** y crea un gráfico comparativo de los tiempos frente a los tamaños de entrada.

- **Parámetros:**
  - **file_paths**: lista de rutas de archivos **.in** que se analizarán.
- **Proceso Interno:**
  1. Inicializa las listas **sizes** y **times** para almacenar los tamaños de entrada y los tiempos de ejecución.
  2. Para cada archivo en **file_paths**:
     - Mide el tiempo antes de la ejecución (**start_time**).
     - Ejecuta **leer_entrada** para leer el archivo y **solution** para resolver los casos de prueba.
     - Mide el tiempo después de la ejecución (**end_time**).
     - Calcula el tiempo transcurrido (**elapsed_time**).
     - Determina el tamaño del archivo en líneas (esto representa el tamaño de la entrada) y a los añade a **sizes**.
     - Añade el tiempo transcurrido a **times**.

3. Genera una gráfica de líneas usando **`matplotlib`**, donde el eje x representa el tamaño de entrada y el eje y el tiempo de ejecución.

```
analyze_complexity(file_paths_in)
```



Observaciones del Gráfico

1. **Relación Lineal**:
   - La línea formada por los puntos muestra un crecimiento lineal en el tiempo de ejecución a medida que aumenta el tamaño del archivo de entrada. Esto indica que el tiempo de procesamiento aumenta de manera proporcional con el tamaño de los datos de entrada.
2. **Escala de Tiempo**:
   - El tiempo de ejecución es bajo (en el rango de milisegundos), lo cual es adecuado para los tamaños de entrada analizados. Sin embargo, esto puede cambiar si el tamaño de entrada crece significativamente, dado que la pendiente sugiere un incremento constante.
3. **Pendiente Constante**:
   - La pendiente de la línea es consistente, lo que sugiere que la complejidad del programa es lineal respecto al tamaño de los datos de entrada. En este contexto, el programa probablemente tiene una complejidad $O(N)$ para estos tamaños de entrada.

Conclusiones

1. **Eficiencia del Programa**:

- Para los tamaños de entrada utilizados, el programa demuestra una eficiencia aceptable y un tiempo de respuesta rápido. Esto es positivo, especialmente si el objetivo es procesar archivos de entrada de tamaño similar.

2. **Escalabilidad**:

   - Aunque el tiempo de ejecución es bajo para los tamaños probados, si el tamaño de entrada continúa creciendo, el tiempo de ejecución también aumentará linealmente. Esto indica que el programa debería manejar bien entradas moderadamente grandes, pero puede volverse más lento en el caso de archivos extremadamente grandes.

En general, el gráfico sugiere que el programa tiene un comportamiento predecible y escalable en un contexto de tamaño de entrada moderado, y su eficiencia es adecuada para los tamaños de datos probados.