



**Facultad de
Ciencias**
UNAM

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

**Práctica 2: Algoritmos de creación y verificación de certificados
para el problema de ruta inducida**

ALUMNOS

Ulises Rodríguez García - 318042202

Zurisadai Uribe García - 318223197

Javier Alejandro Rivera Zavala - 311288876

PROFESOR

Oscar Hernández Constantino

AYUDANTE

Malinali González Lara

ASIGNATURA

Complejidad computacional

Octubre 18, 2023

Algoritmos de verificación y creación de certificados

Para el problema de **RUTA INDUCIDA**

- Prueba que el problema pertenece a NP .

Para demostrar que el problema pertenece a NP , es suficiente mostrar que una solución candidata (o certificado) puede ser verificada en tiempo polinómico y que la generación de un certificado también se realiza en tiempo polinómico.

Certificado:

Dado un ejemplar del problema de la ruta inducida $G = \{V, E\}$ y $0 < k \leq |V|$, un certificado para dicha instancia a del problema, es un subconjunto de vértices $V' = v_1, v_2, \dots, v_{|V'|}$ de la gráfica G del ejemplar, que cumple con:

1. $|V'| = k$.
2. Los vértices de V' inducen una subgráfica que es una ruta simple en G .

Generación de Certificado:

1. Tomamos k y el número de vértices de n de G .
2. Verificamos que $0 < k \leq n$
3. Generamos aleatoriamente k números distintos entre 0 y $n-1$ que representarán índices de los vértices en la secuencia. Esto se hace en tiempo $O(n)$.
4. Creamos una cadena (o lista) de n ceros en tiempo $O(n)$
5. Basándonos en los índices indicados por los números aleatorios, recorremos la cadena y cambiamos ceros por unos en las posiciones correspondientes. Esto es $O(n)$.

El tiempo total de generación del certificado es $O(n)$.

Verificación del certificado:

El proceso de verificación para asegurarse de que el certificado es una solución válida involucra:

1. Comprobar que la secuencia tiene al menos K vértices. Esta verificación es directa y se realiza en $O(|V|)$ tiempo.
2. Confirmar que los vértices de la secuencia forman una ruta simple en G . Para esto, debemos:
 - Asegurarnos que cada par de vértices consecutivos v_i y v_{i+1} están conectados por una arista en E . Recorriendo la lista, esto se logra en $O(|V'|)$ tiempo.
 - Garantizar que no hay vértices repetidos en la secuencia. Utilizando una estructura de datos, como un conjunto, podemos lograr esto en $O(|V'|)$ tiempo.
 - Verificar que no existen aristas adicionales entre los vértices de V' aparte de las definidas por la secuencia. Comprobando todas las posibles combinaciones de vértices en V' , esta verificación toma $O(|V'|^2)$ tiempo.

Análisis:

El proceso de verificación de una solución candidata (certificado) es polinómico en función del tamaño de la entrada, específicamente $O(|V'|^2)$. Asimismo, el proceso de generación de un certificado también es polinómico, siendo $O(n)$. Por lo tanto, podemos afirmar que el problema de RUTA INDUCIDA pertenece a la clase NP .

Para más claridad y detalles sobre cómo generar y verificar los certificados, consultar nuestros algoritmos propuestos. En dichos algoritmos se aprecia que la complejidad es polinomial, para la generación de certificados sólo requerimos generar una lista de n enteros entre 0 y $n-1$, revolverla aleatoriamente (toma tiempo lineal) y luego tomar los k primeros elementos (tiempo constante). Luego de eso recorremos una vez una lista de dimensión n para cambiar ceros por unos de ser preciso lo cuál también toma tiempo lineal.

En el caso de la verificación del certificado, se efectúan una serie de verificaciones por separado que toman tiempo constante, lineal o toman tiempo cuadrático a lo más, esto por que requieren comparar valores directos, iterar sobre listas de dimensión igual al número de vértices (alguna con k elementos donde k es

menor o igual al número de vértices) o bien iterar sobre un conjunto de filas de una matriz y cada una de sus casillas, que en el peor caso pueden ser todas las filas de una matriz $n \times n$, donde n es el número de vértices de la gráfica que representan.

- Iterar sobre el certificado para contar unos: $O(n)$ (donde n es el número de vértices de G).
- Iterar sobre los vértices en el certificado y sus vecinos para verificar grados: $O(n^2)$ (peor caso, ya que iteras sobre todos los pares de vértices).
- Verificación Final de Hojas: $O(1)$ (operaciones constantes).
- Hallar los vecinos de cada hoja: $O(n)$ para cada uno pues requiere de iterar sobre una fila de la matriz.
- Algoritmo para buscar vecinos y formar la trayectoria: $O(n^2)$, en el peor caso debido al bucle while, los for internos iteran sobre filas lo cuál toma tiempo lineal n y el bucle while puede requerir de efectuarse a lo más para las n filas.
- Verificación del contador y otras que no mencionamos: toman tiempo constante pues sólo se comparan valores.

En resumen, nuestro algoritmo de verificación toma tiempo polinomial y con ello exhibimos también que el problema está en NP.

■ Descripción del certificado y esquema de codificación a utilizar

Certificado: Para manejar los ejemplares del problema, emplearemos el mismo esquema de codificación de nuestra práctica anterior e incluso utilizaremos su código para el programa que se nos pide.

Un certificado es un subconjunto de vértices de G tal que el número de elementos que lo constituyen es menor o igual al número de vértices de G , en particular igual al parámetro k . No aceptaremos $k = 0$ pues tal magnitud no supone necesaria ninguna creación o verificación de certificado, pues no tiene sentido en el contexto del problema y tampoco aceptaremos vértices repetidos. La representación del certificado se dará en términos de una cadena binaria de longitud n , donde n es el número de vértices de la gráfica G . En esta cadena:

1. Un “1” en la posición i indica que el vértice correspondiente al índice i es parte del subconjunto certificado.
2. Un “0” en la posición i significa que el vértice correspondiente al índice i no es parte del subconjunto certificado.

1. **Generación del certificado:** Emplearemos el programa y el esquema de codificación de la práctica anterior para generar representaciones de ejemplares del problema de la ruta inducida, sobre tal base procederemos como sigue

- a) Dados k y n (donde n es el número de vértices en G y puede ser deducido del número de filas/columnas de la matriz de adyacencia), se verifica que k esté en el rango $1 \leq k \leq n$.
- b) Se generan aleatoriamente k números distintos entre 0 y $n - 1$. Estos números representan los índices de los vértices que serán parte del certificado.
- c) Se crea una cadena de n ceros. Al recorrer la cadena, si el índice de la posición actual corresponde a alguno de los k números aleatorios, se cambia el “0” por un “1”.

2. **Verificación del certificado:**

- a) Se verifica que la longitud del certificado sea n , que corresponde al número de vértices de G .
- b) Se asegura que el certificado contenga exactamente k unos y que el certificado solo contenga “0s” y “1s”.
- c) Si $k = 1$, se acepta el certificado, pues es una solución trivial.
- d) Si $k \geq 2$, se recorren las filas de la matriz de adyacencia correspondientes a los índices donde el certificado tiene un “1”:
 - 1) Se verifica que exactamente dos de esas filas representen vértices con un grado de 1 respecto a los vértices indicados en el certificado (son adyacentes a un sólo vértice dentro del certificado).
 - 2) Las filas restantes deben representar vértices que tengan grado 2 respecto a los vértices indicados en el certificado.

e) Si no se cumple alguna de las condiciones anteriores, rechazamos, en caso contrario procedemos como sigue: tomamos las hojas como vértices de referencia y si $k = 2$, aceptamos el certificado como válido pues ya sabemos en este punto que son vecinos entre sí, de no serlo, ya habríamos rechazado el certificado. Si $k > 2$ entonces buscamos los vecinos de las hojas y si estos son las hojas mismas (las hojas son vecinos entre sí) entonces rechazamos pues en este punto ya sabemos que hay algún vértice más y las hojas no están conectadas a esos otros vértices. Una vez hallados a los vecinos de las hojas, verificamos si son el mismo, de ser así y si $k=3$ aceptamos, de lo contrario proseguimos.

Tomando a los vecinos de los anteriores vértices (en principio son las hojas) como referencia, buscamos en la matriz a su otro vecino dentro del subconjunto elegido (que no es el que nos sirvió de referencia antes), para cada uno de los vértices de referencia, si es el mismo vecino para ambos vértices de referencia o bien, si un vértice de referencia ya es vecino del otro, también terminamos. Mientras no se cumpla una de las condiciones de paro anterior, proseguiremos actualizando los vértices de referencia tomando como tales a los nuevos vecinos encontrados para los anteriores vértices de referencia.

Durante todo este proceso, llevaremos un contador de los vértices usados como referencia y si al final el total de vértices recorridos es igual a k , sabremos que todos los vértices del subconjunto forman parte de una subgráfica inducida que es una trayectoria de longitud k y por lo tanto la aceptamos como válida.

Pseudocódigo del algoritmo de verificación

Entrada: Matriz de adyacencia A de la gráfica G , certificado C , valor k

Salida: Verdadero si el certificado es válido, Falso en caso contrario

```

1  FUNCION VerificarCertificado(A, C, k) -> bool:
2      n = longitud(A)
3      si longitud(C) != n:
4          retornar Falso
5
6      contarUnos = contar ocurrencias de "1" en C
7      si contarUnos != k:
8          retornar Falso
9
10     si k = 1:
11         retornar Verdadero
12
13     verticesConGrado1 = 0
14     para i = 0 hasta n-1:
15         si C[i] = "1":
16             grado = 0
17             para j = 0 hasta n-1:
18                 si C[j] = "1" y A[i][j] = 1:
19                     grado = grado + 1
20
21             si grado = 1:
22                 verticesConGrado1 = verticesConGrado1 + 1
23             sino si grado != 2:
24                 retornar Falso
25
26     si verticesConGrado1 != 2:
27         retornar Falso
28     si k == 2:
29         retornar Verdadero
30     listaVertCerti = [-1]*k
31     listaVertCerti[0] = hojas[0]
32     listaVertCerti[-1] = hojas[1]
33
34     vecino1 = obtenerVecino(hoja1)
35     vecino2 = obtenerVecino(hoja2)
36
37     si vecino1 == vecinoHoja2 y k==3:
38         retornar Verdadero
39
40     si vecino1 == hoja2 y vecinoHoja2 == hoja1:

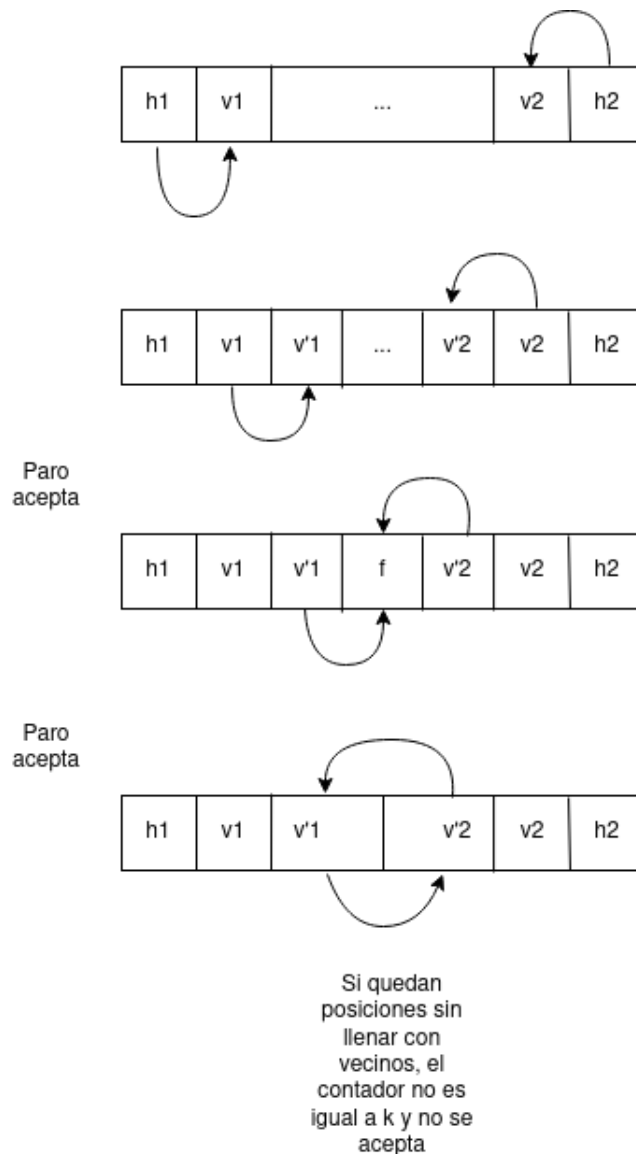
```

```

41     retornar Falso
42
43     listaVertCerti[1] = vecino1
44     listaVertCerti[-2] = vecino2
45
46     i = 0
47     contador = 4
48     conteo = sigueTrayectoria(A, C, contador, vecino1, vecino2)
49     si conteo != k
50         retornar Falso
51     retornar True

```

obtenerVecino() es el bucle que busca el vecino dentro del subconjunto que no es aquel que ya incluimos en la lista, sigueTrayectoria() es el bucle que añade a la listaVertCerti a los vecinos hallados, cada uno al lado de su vecino. El bucle sigueTrayectoria() sigue añadiendo los vértices a la lista siguiendo el patrón que se muestra a continuación y lleva el conteo de los vértices añadidos, para ello usa a obtenerVecino() para buscar los extremos correspondientes:



- Resumen de las pruebas ejecutadas, indicando el ejemplar y certificado usado, así como capturas de pantalla que muestren los resultados de la ejecución.
- Ejemplar con $|V| = 9$, $|E| = 10$, $K = 3$.

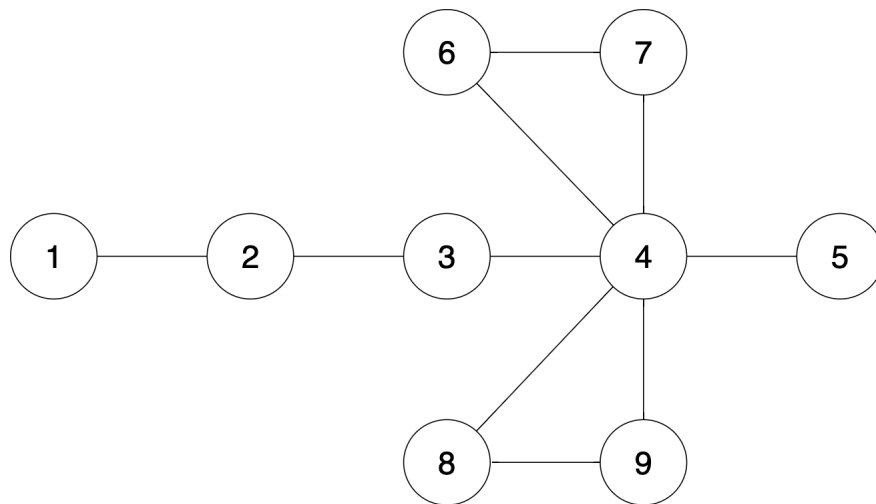
Cadena que codifica a cada ejemplar

La cadena que codifica el ejemplar con un grafo que tiene 9 vértices y 11 aristas, con $K=3$, se presenta a continuación:

001000011110010100000001010000000101000000010111110001000010001001000000101000000100001000110010

Representación Visual del Ejemplar (sin codificar)

A continuación, se muestra una representación visual del grafo sin codificar:



Generación del certificado

- Primer certificado

Procedemos a la generación del certificado para el ejemplar mencionado:

```

Terminal
at ~/Downloads via v3.9.6
> python3 CertificadoYVerificacion.py

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: 1
Nombre del archivo de entrada: ejemplar1.txt
Nombre del archivo de salida: certificado1-1.txt
Certificado escrito en el archivo certificado1-1.txt


Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: |
  
```

El resultado de esta generación es un archivo .txt que contiene el siguiente certificado:

011100000

- Segundo certificado

Procedemos a la generación del certificado para el ejemplar mencionado:



```
Terminal
at ~/Downloads via v3.9.6
> python3 CertificadoYVerificacion.py

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: 1
Nombre del archivo de entrada: ejemplar1.txt
Nombre del archivo de salida: certificado1-2.txt
Certificado escrito en el archivo certificado1-2.txt

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: |
```

El resultado de esta generación es un archivo .txt que contiene el siguiente certificado:

001100001

- Tercer certificado

Procedemos a la generación del certificado para el ejemplar mencionado:



```
Terminal
at ~/Downloads via v3.9.6
> python3 CertificadoYVerificacion.py

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: 1
Nombre del archivo de entrada: ejemplar1.txt
Nombre del archivo de salida: certificado1-3.txt
Certificado escrito en el archivo certificado1-3.txt

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: |
```

El resultado de esta generación es un archivo .txt que contiene el siguiente certificado:

001000110

- Cuarto certificado

Procedemos a la generación del certificado para el ejemplar mencionado:



```
Terminal
at ~/Downloads via v3.9.6
> python3 CertificadoYVerificacion.py

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: 1
Nombre del archivo de entrada: ejemplar1.txt
Nombre del archivo de salida: certificado1-4.txt
Certificado escrito en el archivo certificado1-4.txt

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: |
```

El resultado de esta generación es un archivo .txt que contiene el siguiente certificado:

100101000

- Quinto certificado

Procedemos a la generación del certificado para el ejemplar mencionado:



```
Terminal
at ~/Downloads via v3.9.6
> python3 CertificadoYVerificacion.py

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: 1
Nombre del archivo de entrada: ejemplar1.txt
Nombre del archivo de salida: certificado1-5.txt
Certificado escrito en el archivo certificado1-5.txt

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: |
```

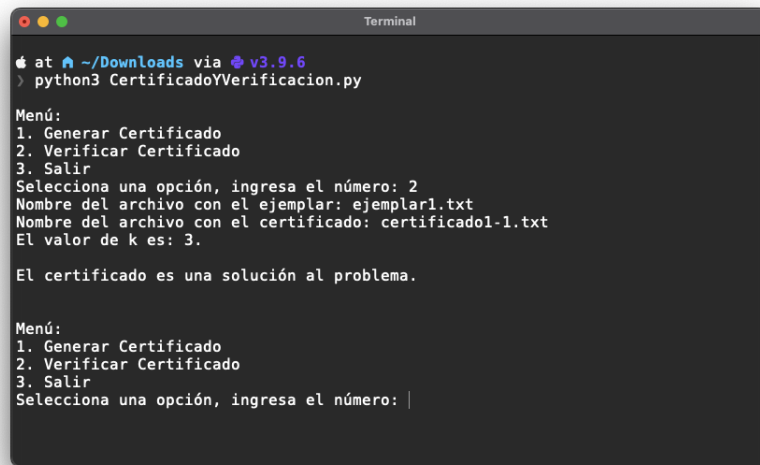
El resultado de esta generación es un archivo .txt que contiene el siguiente certificado:

001000011

Verificación del certificado

- Primera verificación

A continuación, verificamos la validez del primer certificado previamente generado:



```
Terminal
at ~/Downloads via v3.9.6
> python3 CertificadoYVerificacion.py

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: 2
Nombre del archivo con el ejemplar: ejemplar1.txt
Nombre del archivo con el certificado: certificado1-1.txt
El valor de k es: 3.

El certificado es una solución al problema.

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: |
```

Tras la verificación, podemos confirmar que el certificado proporcionado es válido para el ejemplar en cuestión. Esto significa que el certificado cumple con las propiedades y restricciones del problema, y representa una solución correcta para la instancia dada.

- Segunda verificación

A continuación, verificamos la validez del segundo certificado previamente generado:



```
Terminal
at ~/Downloads via v3.9.6
> python3 CertificadoYVerificacion.py

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: 2
Nombre del archivo con el ejemplar: ejemplar1.txt
Nombre del archivo con el certificado: certificado1-2.txt
El valor de k es: 3.

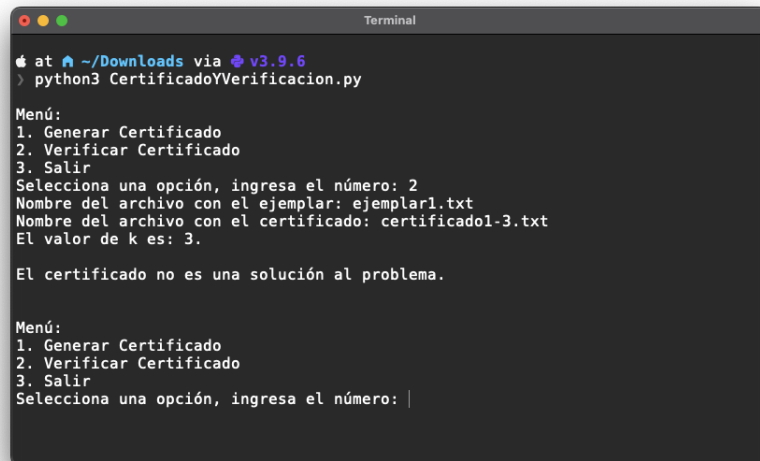
El certificado es una solución al problema.

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: |
```

Tras la verificación, podemos confirmar que el certificado proporcionado es válido para el ejemplar en cuestión. Esto significa que el certificado cumple con las propiedades y restricciones del problema, y representa una solución correcta para la instancia dada.

- o Tercer verificación

A continuación, verificamos la validez del tercer certificado previamente generado:



```
Terminal
at ~/Downloads via v3.9.6
> python3 CertificadoYVerificacion.py

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: 2
Nombre del archivo con el ejemplar: ejemplar1.txt
Nombre del archivo con el certificado: certificado1-3.txt
El valor de k es: 3.


El certificado no es una solución al problema.

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: |
```

Según los resultados obtenidos, el certificado proporcionado no es válido para el ejemplar en cuestión.

- o Cuarta verificación

A continuación, verificamos la validez del cuarto certificado previamente generado:



```
Terminal
at ~/Downloads via v3.9.6
> python3 CertificadoYVerificacion.py

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: 2
Nombre del archivo con el ejemplar: ejemplar1.txt
Nombre del archivo con el certificado: certificado1-4.txt
El valor de k es: 3.

El certificado no es una solución al problema.

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: |
```

Según los resultados obtenidos, el certificado proporcionado no es válido para el ejemplar en cuestión.

- Quinta verificación

A continuación, verificamos la validez del quinto certificado previamente generado:

```

Terminal
🍏 at ~/Downloads via v3.9.6
> python3 CertificadoYVerificacion.py

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: 2
Nombre del archivo con el ejemplar: ejemplar1.txt
Nombre del archivo con el certificado: certificado1-5.txt
El valor de k es: 3.

El certificado no es una solución al problema.

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: |

```

Según los resultados obtenidos, el certificado proporcionado no es válido para el ejemplar en cuestión.

- Ejemplar con $|V| = 8$, $|E| = 10$, $K = 4$.

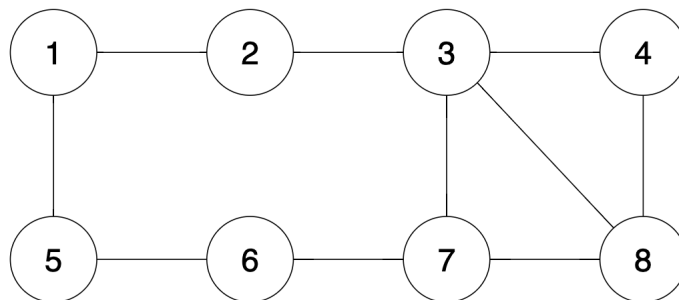
Cadena que codifica a cada ejemplar

La representación codificada del ejemplar que describe un grafo con 8 vértices y 10 aristas, con $K=4$, se presenta a continuación:

00010000110010000100100010100000010100110010000110000100000010100010010100110010

Representación Visual del Ejemplar (sin codificar)


A continuación, se muestra una representación visual del grafo sin codificar:



Generación del certificado

- Primer certificado

Procedemos a la generación del certificado para el ejemplar mencionado:



```
Terminal
at ~/Downloads via v3.9.6
> python3 CertificadoYVerificacion.py

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: 1
Nombre del archivo de entrada: ejemplar2.txt
Nombre del archivo de salida: certificado2-1.txt
Certificado escrito en el archivo certificado2-1.txt

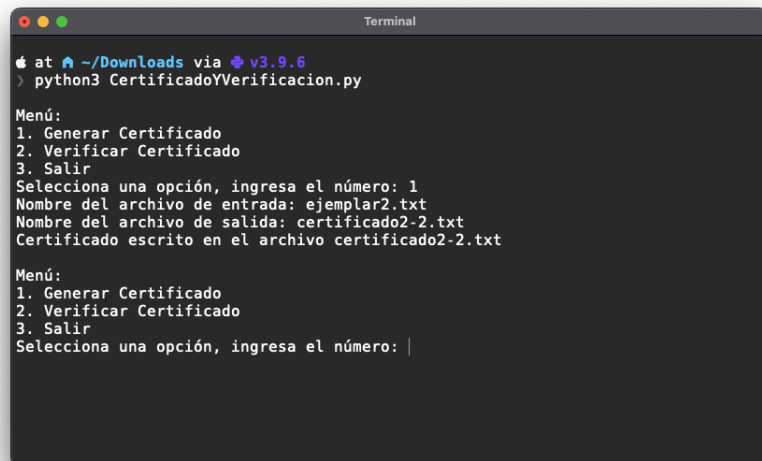
Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: |
```

El resultado de esta generación es un archivo .txt que contiene el siguiente certificado:

01001110

- Segundo certificado

Procedemos a la generación del certificado para el ejemplar mencionado:



```
Terminal
at ~/Downloads via v3.9.6
> python3 CertificadoYVerificacion.py

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: 1
Nombre del archivo de entrada: ejemplar2.txt
Nombre del archivo de salida: certificado2-2.txt
Certificado escrito en el archivo certificado2-2.txt

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: |
```

El resultado de esta generación es un archivo .txt que contiene el siguiente certificado:

10101001

- o Tercer certificado

Procedemos a la generación del certificado para el ejemplar mencionado:



```
Terminal
at ~/Downloads via v3.9.6
> python3 CertificadoYVerificacion.py

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: 1
Nombre del archivo de entrada: ejemplar2.txt
Nombre del archivo de salida: certificado2-3.txt
Certificado escrito en el archivo certificado2-3.txt

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: |
```

El resultado de esta generación es un archivo .txt que contiene el siguiente certificado:

11001001

- o Cuarto certificado

Procedemos a la generación del certificado para el ejemplar mencionado:



```
Terminal
at ~/Downloads via v3.9.6
> python3 CertificadoYVerificacion.py

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: 1
Nombre del archivo de entrada: ejemplar2.txt
Nombre del archivo de salida: certificado2-4.txt
Certificado escrito en el archivo certificado2-4.txt

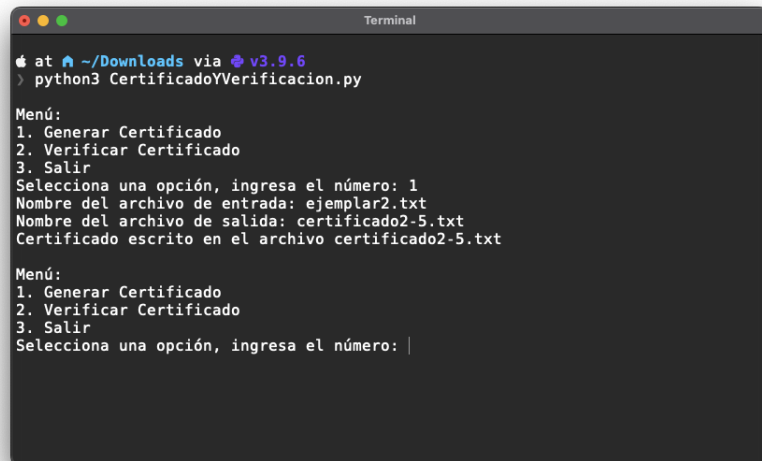
Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: |
```

El resultado de esta generación es un archivo .txt que contiene el siguiente certificado:

01011010

- Quinto certificado

Procedemos a la generación del certificado para el ejemplar mencionado:



```
Terminal
at ~/Downloads via v3.9.6
> python3 CertificadoYVerificacion.py

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: 1
Nombre del archivo de entrada: ejemplar2.txt
Nombre del archivo de salida: certificado2-5.txt
Certificado escrito en el archivo certificado2-5.txt

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: |
```

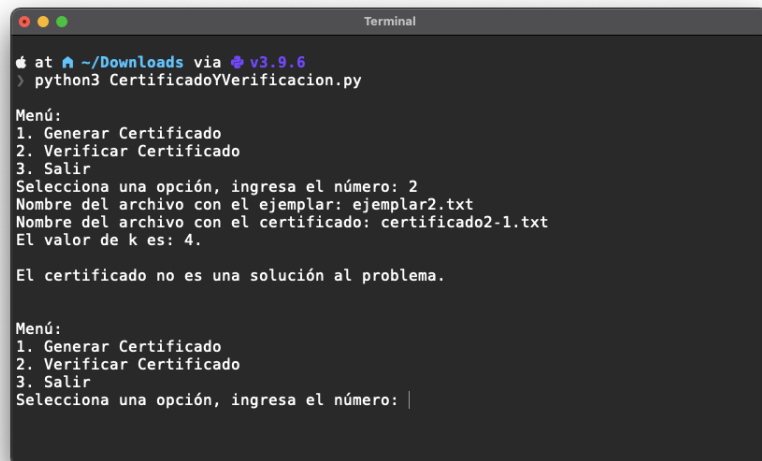
El resultado de esta generación es un archivo .txt que contiene el siguiente certificado:

01010101

Verificación del certificado

- Primer verificación

A continuación, verificamos la validez del primer certificado previamente generado:



```
Terminal
at ~/Downloads via v3.9.6
> python3 CertificadoYVerificacion.py

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: 2
Nombre del archivo con el ejemplar: ejemplar2.txt
Nombre del archivo con el certificado: certificado2-1.txt
El valor de k es: 4.

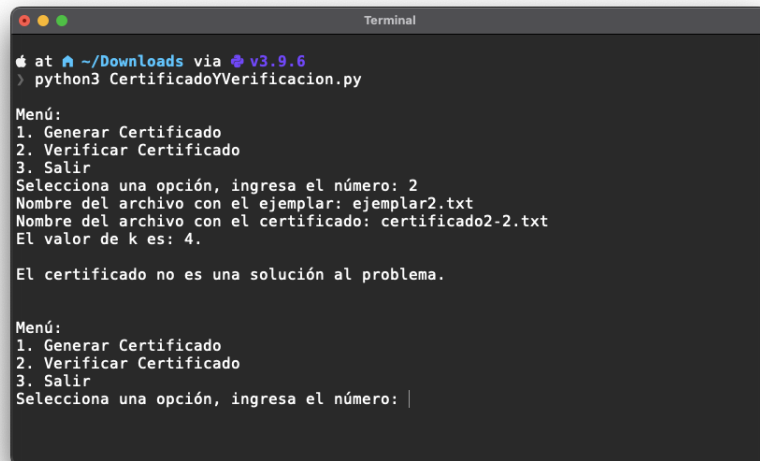
El certificado no es una solución al problema.

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: |
```

Según los resultados obtenidos, el certificado proporcionado no es válido para el ejemplar en cuestión.

- Segunda verificación

A continuación, verificamos la validez del segundo certificado previamente generado:



```
Terminal
Apple at ~ /Downloads via v3.9.6
> python3 CertificadoYVerificacion.py

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: 2
Nombre del archivo con el ejemplar: ejemplar2.txt
Nombre del archivo con el certificado: certificado2-2.txt
El valor de k es: 4.

El certificado no es una solución al problema.

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: |
```

Según los resultados obtenidos, el certificado proporcionado no es válido para el ejemplar en cuestión.

- Tercer verificación

A continuación, verificamos la validez del tercer certificado previamente generado:



```
Terminal
Apple at ~ /Downloads via v3.9.6
> python3 CertificadoYVerificacion.py

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: 2
Nombre del archivo con el ejemplar: ejemplar2.txt
Nombre del archivo con el certificado: certificado2-3.txt
El valor de k es: 4.

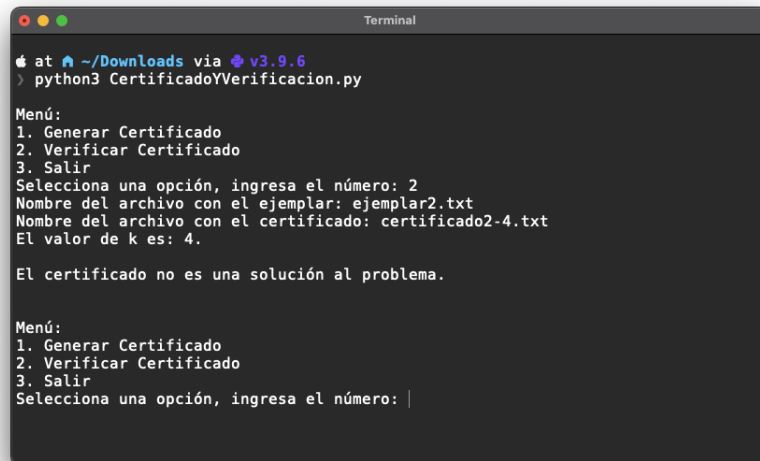
El certificado no es una solución al problema.

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: |
```

Según los resultados obtenidos, el certificado proporcionado no es válido para el ejemplar en cuestión.

- Cuarta verificación

A continuación, verificamos la validez del cuarto certificado previamente generado:



```
Terminal
Apple at ~/Downloads via v3.9.6
> python3 CertificadoYVerificacion.py

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: 2
Nombre del archivo con el ejemplar: ejemplar2.txt
Nombre del archivo con el certificado: certificado2-4.txt
El valor de k es: 4.

El certificado no es una solución al problema.

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: |
```

Según los resultados obtenidos, el certificado proporcionado no es válido para el ejemplar en cuestión.

- Quinto verificación

A continuación, verificamos la validez del quinto certificado previamente generado:



```
Terminal
Apple at ~/Downloads via v3.9.6
> python3 CertificadoYVerificacion.py

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: 2
Nombre del archivo con el ejemplar: ejemplar2.txt
Nombre del archivo con el certificado: certificado2-5.txt
El valor de k es: 4.

El certificado no es una solución al problema.

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: |
```

Según los resultados obtenidos, el certificado proporcionado no es válido para el ejemplar en cuestión.

- Ejemplar con $|V| = 5$, $|E| = 8$, $K = 2$.

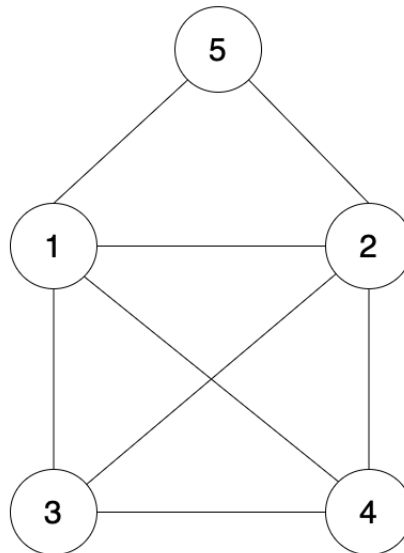
Cadena que codifica a cada ejemplar

La representación codificada del ejemplar que describe un grafo con 5 vértices y 8 aristas, con $K=2$, se presenta a continuación:

0010001101010111110111110101110011000

Representación Visual del Ejemplar (sin codificar)

A continuación, se muestra una representación visual del grafo sin codificar:



Generación del certificado

- Primer certificado

Procedemos a la generación del certificado para el ejemplar mencionado:

```

Terminal
at ~/Downloads via v3.9.6
python3 CertificadoYVerificacion.py

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: 1
Nombre del archivo de entrada: ejemplar3.txt
Nombre del archivo de salida: certificado3-1.txt
Certificado escrito en el archivo certificado3-1.txt

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: |
  
```

El resultado de esta generación es un archivo .txt que contiene el siguiente certificado:

00011

- Segundo certificado

Procedemos a la generación del certificado para el ejemplar mencionado:



```
Terminal
at ~/Downloads via v3.9.6
> python3 CertificadoYVerificacion.py

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: 1
Nombre del archivo de entrada: ejemplar3.txt
Nombre del archivo de salida: certificado3-2.txt
Certificado escrito en el archivo certificado3-2.txt

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: |
```

El resultado de esta generación es un archivo .txt que contiene el siguiente certificado:

01100

- Tercer certificado

Procedemos a la generación del certificado para el ejemplar mencionado:



```
Terminal
at ~/Downloads via v3.9.6
> python3 CertificadoYVerificacion.py

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: 1
Nombre del archivo de entrada: ejemplar3.txt
Nombre del archivo de salida: certificado3-3.txt
Certificado escrito en el archivo certificado3-3.txt

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: |
```

El resultado de esta generación es un archivo .txt que contiene el siguiente certificado:

11000

- Cuarto certificado

Procedemos a la generación del certificado para el ejemplar mencionado:



```
Terminal
at ~/Downloads via v3.9.6
> python3 CertificadoYVerificacion.py

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: 1
Nombre del archivo de entrada: ejemplar3.txt
Nombre del archivo de salida: certificado3-4.txt
Certificado escrito en el archivo certificado3-4.txt

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: |
```

El resultado de esta generación es un archivo .txt que contiene el siguiente certificado:

10010

- Quinto certificado

Procedemos a la generación del certificado para el ejemplar mencionado:



```
Terminal
at ~/Downloads via v3.9.6
> python3 CertificadoYVerificacion.py

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: 1
Nombre del archivo de entrada: ejemplar3.txt
Nombre del archivo de salida: certificado3-5.txt
Certificado escrito en el archivo certificado3-5.txt

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: |
```

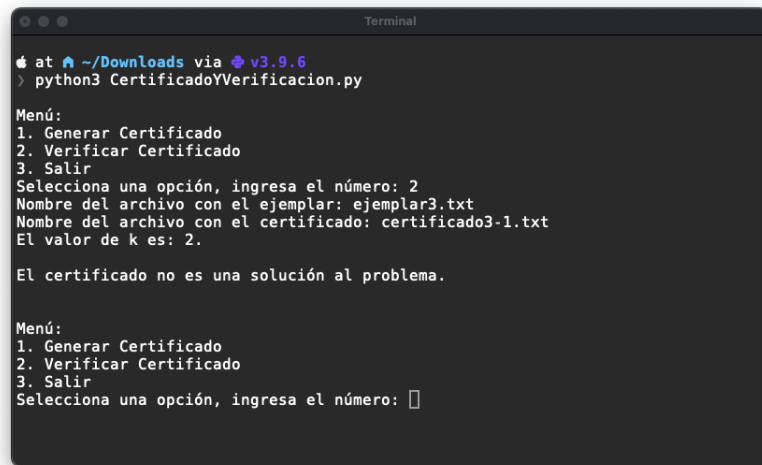
El resultado de esta generación es un archivo .txt que contiene el siguiente certificado:

10100

Verificación del certificado

- Primera verificación

A continuación, verificamos la validez del primer certificado previamente generado:



```
Terminal
at ~/Downloads via v3.9.6
> python3 CertificadoYVerificacion.py

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: 2
Nombre del archivo con el ejemplar: ejemplar3.txt
Nombre del archivo con el certificado: certificado3-1.txt
El valor de k es: 2.

El certificado no es una solución al problema.

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: 
```

Según los resultados obtenidos, el certificado proporcionado no es válido para el ejemplar en cuestión.

- Segunda verificación

A continuación, verificamos la validez del segundo certificado previamente generado:



```
Terminal
at ~/Downloads via v3.9.6
> python3 CertificadoYVerificacion.py

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: 2
Nombre del archivo con el ejemplar: ejemplar3.txt
Nombre del archivo con el certificado: certificado3-2.txt
El valor de k es: 2.

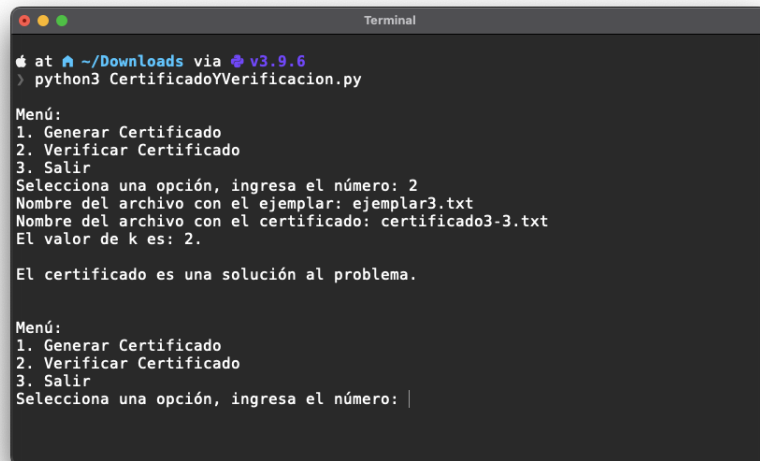
El certificado es una solución al problema.

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: |
```

Tras la verificación, podemos confirmar que el certificado proporcionado es válido para el ejemplar en cuestión. Esto significa que el certificado cumple con las propiedades y restricciones del problema, y representa una solución correcta para la instancia dada.

- Tercera verificación

A continuación, verificamos la validez del tercer certificado previamente generado:



```
Terminal
at ~/Downloads via v3.9.6
> python3 CertificadoYVerificacion.py

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: 2
Nombre del archivo con el ejemplar: ejemplar3.txt
Nombre del archivo con el certificado: certificado3-3.txt
El valor de k es: 2.


El certificado es una solución al problema.

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: |
```

Tras la verificación, podemos confirmar que el certificado proporcionado es válido para el ejemplar en cuestión. Esto significa que el certificado cumple con las propiedades y restricciones del problema, y representa una solución correcta para la instancia dada.

- Cuarta verificación

A continuación, verificamos la validez del cuarto certificado previamente generado:



```
Terminal
at ~/Downloads via v3.9.6
> python3 CertificadoYVerificacion.py

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: 2
Nombre del archivo con el ejemplar: ejemplar3.txt
Nombre del archivo con el certificado: certificado3-4.txt
El valor de k es: 2.

El certificado es una solución al problema.

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: |
```

Tras la verificación, podemos confirmar que el certificado proporcionado es válido para el ejemplar en cuestión. Esto significa que el certificado cumple con las propiedades y restricciones del problema, y representa una solución correcta para la instancia dada.

- Quinta verificación

A continuación, verificamos la validez del quinto certificado previamente generado:



```
Terminal
at ~/Downloads via v3.9.6
> python3 CertificadoYVerificacion.py

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: 2
Nombre del archivo con el ejemplar: ejemplar3.txt
Nombre del archivo con el certificado: certificado3-5.txt
El valor de k es: 2.

El certificado es una solución al problema.

Menú:
1. Generar Certificado
2. Verificar Certificado
3. Salir
Selecciona una opción, ingresa el número: |
```

Tras la verificación, podemos confirmar que el certificado proporcionado es válido para el ejemplar en cuestión. Esto significa que el certificado cumple con las propiedades y restricciones del problema, y representa una solución correcta para la instancia dada.

Conclusión:

A través de las pruebas realizadas, hemos sometido a diversos ejemplares al proceso de generación y verificación de certificados. Estos ejemplares, representando instancias del problema de RUTA INDUCIDA, presentaron grafo con características variadas, en términos de cantidad de vértices, aristas y el valor de K .

La cadena codificada proporcionó un medio eficiente para representar el grafo, permitiendo una comprensión clara y concisa del problema. Además, las representaciones visuales ofrecieron una percepción gráfica del escenario, facilitando la interpretación y el análisis.

La generación del certificado y su posterior verificación sirvieron para comprobar la validez y aplicabilidad de la solución propuesta. Mientras algunos certificados resultaron ser válidos, indicando una solución correcta al problema, otros no cumplieron con las restricciones y propiedades requeridas.

En conjunto, este ejercicio demostró la complejidad inherente al problema de RUTA INDUCIDA, y la importancia de contar con mecanismos eficientes tanto para la representación como para la solución del problema. Las pruebas y verificaciones son herramientas esenciales en la búsqueda de soluciones robustas y precisas en el ámbito de la computación y la teoría de grafos.

Referencias

Prácticamente sólo utilizamos lo visto en clase, adicional a ello sólo consultamos escuetamente estos materiales:

- Graphs and digraphs
Autores: Gary Chartrand, Linda Lezniak y Ping Zhang
Editorial: Taylor & Francis group - CRC press
Año de impresión: 2016
Idioma: Inglés.
Se consultaron los capítulos 1 y 2: *Introduction* en el apartado de *Induced graphs* y *Conected graphs and distance*.
- Notas del curso de análisis de algoritmos 2023-2
Presentación 2 - Notación asintótica
Autor: Pedro Ulises Cervantes González
Link: