

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

Complejidad Computacional 2024-1



PRÁCTICA 1

PROFESOR:

Oscar Hernández Constantino

AYUDANTE:

Malinali González Lara

INTEGRANTES:

Manjarrez Angeles Valeria Fernanda - 317234785

Pedro Mendez Jose Manuel - 315073120

Sánchez Reza Neider - 317020931

19 de septiembre de 2023

Índice

1. Descripción del problema como problema de optimización	2
2. Descripción del problema como problema de decisión	2
3. Esquema de codificación basado en texto	2
4. Esquema de codificación basado en matrices	3
4.1. Algoritmo	3
4.2. Análisis de complejidad	4
5. Ejemplos de ejecución	5
5.1. Ejemplar 1	5
5.1.1. Representación visual de ejemplar	5
5.1.2. Representación del ejemplar con esquema basado en texto	5
5.1.3. Representación del ejemplar con esquema basado en matrices	6
5.2. Ejemplar 2	6
5.2.1. Representación visual de ejemplar	6
5.2.2. Representación del ejemplar con esquema basado en texto	7
5.2.3. Representación del ejemplar con el esquema basado en matrices	7
5.3. Ejemplar 3	8
5.3.1. Representación visual de ejemplar	8
5.3.2. Representación del ejemplar con esquema basado en texto	8
5.3.3. Representación del ejemplar con esquema basado en matrices	9

1. Descripción del problema como problema de optimización

ruta INDUCIDA

- EJEMPLAR: Una gráfica $G = (V, E)$.
- PREGUNTA: ¿Cuál es el subconjunto $V' \subseteq V$ tal que la subgráfica inducida por V' es la ruta simple $|V'|$ con mayor número de vértices?

2. Descripción del problema como problema de decisión

ruta INDUCIDA

- EJEMPLAR: Una gráfica $G = (V, E)$, un entero positivo $K \leq |V|$.
- PREGUNTA: ¿Existe un subconjunto $V' \subseteq V$, con $|V'| \geq K$, tal que la subgráfica inducida por V' es una ruta simple con $|V'|$ vértices?

3. Esquema de codificación basado en texto

Un ejemplar para nuestro problema consiste de una gráfica $G = (V, E)$ y un entero positivo $K \leq |V|$. Por lo que necesitamos codificar G y K .

Para codificar un ejemplar de nuestro problema primero escribiremos el valor de nuestra K , seguido de un salto de línea.

Después escribiremos todos los vértices de nuestra gráfica en una sola línea, separado por una coma (,), los representaremos con números enteros, no se puede repetir un número entero para la representación de un vértice. Al final agregaremos un salto de línea.

Dado que la gráfica es no dirigida y sin pesos, podemos representar la arista ab y ba con una sola arista, por lo que basta con dar los vértices que conforman las aristas separadas por una coma (,) y un salto de línea tras poner cada par de vértices separados por una coma.

Para ver ejemplos de codificación de ejemplares con este esquema, consulte la sección 5 *Ejemplos de ejecución* más adelante en este mismo reporte.

```
1 4 // El valor de nuestra k, k=4
2 1,2,3,4,5,6 // Todos los vertices que componen nuestra grafica
3 1,2 // Una arista entre los vertices 1 y 2
4 2,3
5 3,4
6 4,5
7 5,6
```

Listing 1: Ejemplo de codificación de un ejemplar arbitrario.

4. Esquema de codificación basado en matrices

4.1. Algoritmo

Dado el esquema de codificación basado en texto anterior, podemos definir el siguiente algoritmo que transforma la codificación de un ejemplar dado, a una basada en matrices. Para esta representación, como las gráficas de entrada son no dirigidas y sin pesos, optamos por utilizar la matriz de adyacencia de una gráfica, donde hay un renglón y una columna por cada vértice de la gráfica y la entrada $M_{ij} = 1$ si la arista $i - j$ existe en la gráfica. En pseudocódigo:

```
1 Algoritmo: CodificacionTextoMatriz
2
3 Parametros:
4     in: una cadena de texto que representa la ruta del archivo de entrada.
5     out: una cadena de texto que representa la ruta del archivo de salida.
6
7 CodificacionTextoMatriz(in, out):
8     // Cada linea leida con nextLine() se lee como String
9
10    valor_k = in.nextLine().asInt()
11
12    vertices = in.nextLine()
13    lista_vertices = vertices.split(",")    // Arreglo de cadenas
14    total_vertices = lista_vertices[lista_vertices.length - 1]
15
16    matriz = int[total_vertices][total_vertices]
17
18    total_aristas = 0
19    while (in.hasNext()):
20        arista = in.nextLine()
21        total_aristas++
22        par = arista.split(",")    // Obtiene un arreglo de cadenas
23        origen = par[0].asInt()
24        destino = par[1].asInt()
25
26        matriz[origen][destino] = 1
27        matriz[destino][origen] = 1
28
29    for (int i = 0; i < matriz.length; i++):
30        for (int j = 0; j < matriz.length; j++):
31            out.write(matriz[i][j] + " ")
32        out.write("\n")
33    out.write(valor_k)
34
35    print(total_vertices)
36    print(total_aristas)
37    print(valor_k)
```

A grandes rasgos, el algoritmo lee cada línea del archivo de entrada y lo interpreta según lo establecido en el esquema de codificación basado en texto.

En particular, con el número total de vértices y la codificación en pares de las aristas es posible crear una matriz cuadrada que representa la codificación basada en matriz de la gráfica. Luego de leer cada línea, se escribe en el archivo los datos de la matriz resultante y el valor del parámetro K . Finalmente, se imprimen en consola los valores más importantes del ejemplar.

Nótese que podemos hacer varias suposiciones respecto a la codificación basada en texto:

- La primera línea del esquema de codificación siempre es el valor del parámetro K , por lo que podemos leerlo y guardarlo directamente (línea 10).
- La segunda línea corresponde con la lista de vértices, etiquetados como valores numéricos enteros en orden ascendente empezando en 1, por lo que el total de vértices se obtiene como el último valor de esta lista (línea 14).
- Al ser la gráfica no dirigida y sin pesos y ya que la representación de vértices es numérica, cada arista se representa como 2 enteros separados por una coma. Así, es fácil separarlos y usarlos directamente como índices para llenar la matriz (líneas 22-27).

Para la representación matricial en texto plano, se hace uso de los caracteres de espacio en blanco (para separar entradas de la matriz) y de salto de línea (para separar renglones de la matriz y el parámetro K).

4.2. Análisis de complejidad

Para estimar la complejidad del algoritmo anterior, analizaremos por separado cada etapa en la ejecución del mismo:

- Obtención del valor K : En esta etapa, simplemente se lee una única línea y se almacena el valor obtenido como entero en una variable. El tiempo de ejecución en este caso se mantiene en $O(1)$ con espacio $O(1)$.
- Obtención del total de vértices en la gráfica: En esta etapa se lee una línea para luego obtener el último carácter de la cadena. Según la implementación, la obtención de dicho carácter puede variar según la cantidad de vértices de la gráfica, o ser accedido en tiempo constante, dependiendo de si es posible acceder en tiempo constante a índices de una cadena o no. En el primer caso, la complejidad en tiempo es $O(|V|)$, mientras que en el segundo caso es $O(1)$. En ambos casos, la complejidad en espacio se mantiene en $O(1)$.
- Obtención de aristas y llenado de la matriz: Para esta etapa, es necesario recorrer cada una de las aristas codificadas en el esquema basado en texto. Además, se asume que cada arista tiene el mismo tamaño (en bits), de modo que leer y separar los vértices toma siempre el mismo tiempo, por lo que tenemos en total un tiempo de ejecución de $O(|E|)$. En cuanto a la complejidad en espacio, debido a que es necesario la creación de una matriz cuadrada de tamaño $|V|$, entonces tenemos complejidad de $O(|V|^2)$.
- Escritura de la codificación en archivo: En esta etapa se recorre la matriz creada para escribir su contenido, por lo que el tiempo de ejecución queda $O(|V|^2)$. En este caso, se escribe el contenido de la matriz en un archivo externo, por lo que la complejidad en espacio también es de $O(|V|^2)$.
- Impresión de los valores importantes del ejemplar: Finalmente, sólo es necesario imprimir valores por lo que tanto la complejidad en tiempo como en espacio se mantiene en $O(1)$.

Con lo anterior, podemos concluir que, a grandes rasgos, la complejidad total en tiempo del algoritmo anterior es de $O(|V| + |E| + |V|^2)$ (o bien, $O(|E| + |V|^2)$ si el acceso a índices de cadenas es constante) con complejidad en espacio de $O(|V|^2)$.

5. Ejemplos de ejecución

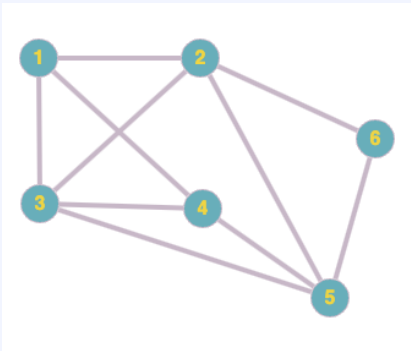
Procedemos a detallar los esquemas de codificación implementados a partir de mostrar los resultados de ejecución con ejemplares concretos.

5.1. Ejemplar 1

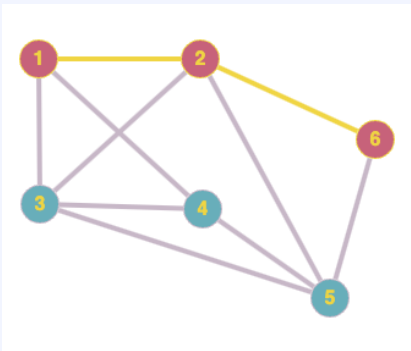
Detallaremos el caso para un ejemplar con $|V| \geq 5, |E| \geq 10, k = 3$ con respuesta SÍ.

5.1.1. Representación visual de ejemplar

Proponemos el siguiente ejemplar que cumple con las características requeridas de que $|V| \geq 5$ y $|E| \geq 10$.



Y podemos notar que SI existe un subconjunto de vértices $V' \subseteq V$ tal que la subgráfica inducida es una ruta simple y el número de vértices de dicha subgráfica es mayor o igual a $k = 3$.



5.1.2. Representación del ejemplar con esquema basado en texto

```
1 3
2 1,2,3,4,5,6
3 1,2
4 1,3
5 1,4
6 2,3
7 2,5
8 2,6
9 3,4
10 3,5
11 4,5
12 5,6
```

Listing 2: Contenido del archivo example1.txt

5.1.3. Representación del ejemplar con esquema basado en matrices

Al ejecutar el algoritmo de transformación de esquema con el ejemplar anterior, se obtiene la siguiente codificación donde la última línea representa el valor del parámetro K :

```

1      0  1  1  1  0  0
2      1  0  1  0  1  1
3      1  1  0  1  1  0
4      1  0  1  0  1  0
5      0  1  1  1  0  1
6      0  1  0  0  1  0
7      3

```

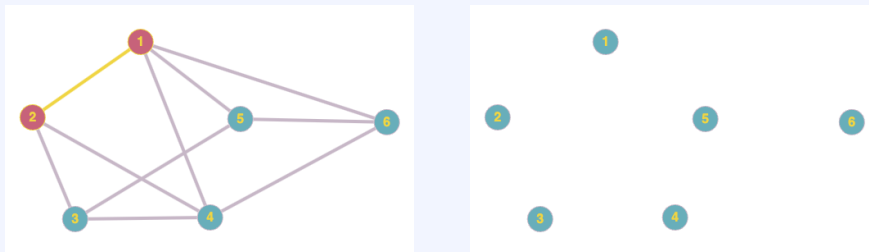
Listing 3: Resultado del programa, en el archivo out_example1.txt

5.2. Ejemplar 2

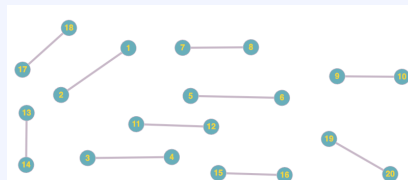
Detallaremos el caso para un ejemplar con $|V| \geq 5$, $|E| \geq 10$, $k = 2$ con respuesta NO.

5.2.1. Representación visual de ejemplar

Proponemos el siguiente ejemplar que cumple con las características requeridas de que $|V| \geq 5$ y $|E| \geq 10$. Sin embargo podemos notar que es imposible que la respuesta sea NO, ya que mientras exista una arista entre dos vertices siempre existirá un subconjunto de vértices $V' \subseteq V$ tal que la subgráfica inducida es una ruta simple y el número de vértices de dicha subgráfica sea mayor o igual a k , en particular mayor o igual a 2. La única manera en que la respuesta sea NO, es que no exista ninguna arista en nuestra gráfica, pero de esta forma no cumplimos con la característica de nuestro ejemplar de que $|E| \geq 10$.



En su lugar detallaremos el caso para un ejemplar con $|V| \geq 5$, $|E| \geq 10$, $k = 3$ con respuesta NO.



Podemos ver que cumple con las características de que $|V| \geq 5$ y $|E| \geq 10$ y a su vez NO existe un conjunto de vertices $V' \subseteq V$ tal que la subgráfica inducida es una ruta simple y el número de vértices de dicha subgráfica sea mayor o igual a 3.

5.2.2. Representación del ejemplar con esquema basado en texto

```

1 3
2 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20
3 1,2
4 3,4
5 5,6
6 7,8
7 9,10
8 11,12
9 13,14
10 15,16
11 17,18
12 19,20

```

Listing 4: Contenido del archivo example2.txt

5.2.3. Representación del ejemplar con el esquema basado en matrices

La siguiente es la representación del ejemplar, una vez ejecutado el programa de transformación de esquemas. Al igual que en el caso anterior, el valor de K se encuentra en la última línea del resultado.

```

1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
3 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
4 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
5 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
6 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
7 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
8 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
9 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
10 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
11 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
12 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
13 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
14 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
15 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
16 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
17 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
18 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
19 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
20 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
21 3

```

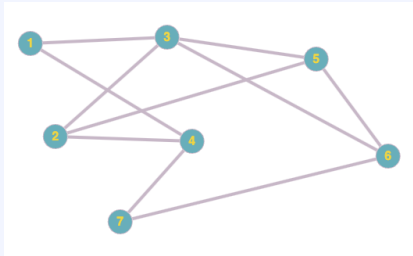
Listing 5: Resultado del programa, en el archivo out_example2.txt

5.3. Ejemplar 3

Detallaremos el caso para un ejemplar con $|V| \geq 6, |E| \geq 10, k = 3$ con respuesta sí.

5.3.1. Representación visual de ejemplar

Proponemos el siguiente ejemplar que cumple con las características requeridas de que $|V| \geq 6$ y $|E| \geq 10$.



Y podemos notar que si existe un subconjunto de vértices $V' \subseteq V$ tal que la subgráfica inducida es una ruta simple y el número de vértices de dicha subgráfica es mayor o igual a $k = 3$.

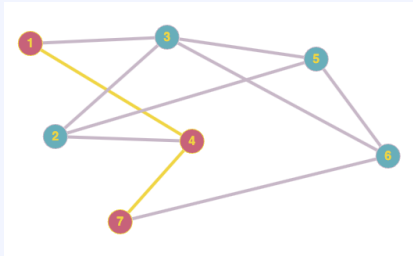


Figura 1: Subgráfica inducida por los vertices 1,4,7

5.3.2. Representación del ejemplar con esquema basado en texto

```
1 3
2 1 , 2 , 3 , 4 , 5 , 6 , 7
3 1 , 3
4 1 , 4
5 2 , 3
6 2 , 4
7 2 , 5
8 3 , 5
9 3 , 6
10 4 , 7
11 5 , 6
12 6 , 7
```

Listing 6: Contenido del archivo example3.txt

5.3.3. Representación del ejemplar con esquema basado en matrices

De manera similar a los ejemplares anteriores, el resultado de la codificación, una vez ejecutado el programa, es la siguiente.

```
1      0 0 1 1 0 0 0
2      0 0 1 1 1 0 0
3      1 1 0 0 1 1 0
4      1 1 0 0 0 0 1
5      0 1 1 0 0 1 0
6      0 0 1 0 1 0 1
7      0 0 0 1 0 1 0
8      3
```

Listing 7: Resultado del programa, en el archivo out_example3.txt