



DEPARTAMENTO  
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

# Algoritmos sobre grafos

Algoritmos y Estructuras de Datos III  
1er cuatrimestre 2022

Integrante	LU	Correo electrónico
Agustin Polleschi	██████	████████████████████
Constanza García	██████	██
Mariano Oca	██████	████████████████████████████████
Lucas Rango	██████	████████████████████████████████
Agustin Gianolini	██████	████████████████████████████████



**Facultad de Ciencias Exactas y Naturales**  
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

<http://www.fcen.uba.ar>

**Palabras Clave:** BFS, Union-Find, algoritmo de Johnson, geodésico, conjunto dominante total, algoritmo de Bertossi

# Índice

<b>1. Introducción</b>	<b>3</b>
<b>2. Ejercicio 1</b>	<b>4</b>
2.1. Descripción del problema . . . . .	4
2.2. El algoritmo . . . . .	4
2.3. Instancias de prueba . . . . .	5
2.4. Resultados . . . . .	5
<b>3. Ejercicio 2</b>	<b>6</b>
3.1. Descripción del problema . . . . .	6
3.2. El algoritmo . . . . .	6
3.3. Instancias de prueba . . . . .	8
3.4. Resultados . . . . .	8
<b>4. Ejercicio 3</b>	<b>10</b>
4.1. Descripción del problema . . . . .	10
4.2. El algoritmo . . . . .	10
4.3. Instancias . . . . .	11
4.4. Resultados . . . . .	11
<b>5. Ejercicio 4</b>	<b>12</b>
5.1. Descripción del problema . . . . .	12
5.2. El algoritmo . . . . .	12
5.3. Instancias de prueba . . . . .	13
5.4. Resultados . . . . .	13
<b>6. Conclusión</b>	<b>14</b>
<b>7. Apéndice</b>	<b>15</b>

# 1. Introducción

El objetivo de este informe es mostrar la puesta a prueba de los conocimientos adquiridos en la materia Algoritmos y Estructura de Datos III en cuanto a grafos y digrafos. Para esto, se realiza la resolución, análisis de complejidad y testeo práctico de los siguientes problemas:

1. Determinar si un grafo conexo es o no **geodésico**.
2. Contar la cantidad de componentes conexas de un grafo representado mediante una matriz  $M \in \{0, 1\}^{m \times n}$ .
3. Hallar el camino mínimo entre todo par de vértices de un grafo conexo y pesado.
4. Hallar un **conjunto dominante total** de una familia de intervalos en  $\mathbb{R}$ .

Para cada uno de estos ejercicios se describirá en detalle el problema, se dará un algoritmo que lo resuelve, se analizará su complejidad temporal (y espacial si lo requiere), se darán instancias de prueba donde se medirá el rendimiento temporal del algoritmo y, finalmente, se compararán los resultados obtenidos con los esperados en base al análisis teórico de complejidad.

## 2. Ejercicio 1

### 2.1. Descripción del problema

Decimos que un grafo es **geodésico** si para todo par de vértices existe un único camino de longitud mínima entre ellos.

El objetivo del problema es, dado un grafo conexo, determinar si es o no geodésico.

### 2.2. El algoritmo

Dado un vértice  $v$ , si hacemos **BFS** enraizado en él obtenemos un árbol que contiene los caminos mínimos desde  $v$  a todos los vértices.

Notemos que, dado  $w$ , si existe una arista  $tw$  fuera del árbol tal que  $d(v, t) + 1 = d(v, w)$ , entonces hay dos caminos de  $v$  a  $w$  con longitud mínima. Esto es porque el camino de  $v$  a  $t$  junto con la arista  $tw$  es un camino de  $v$  a  $w$ .

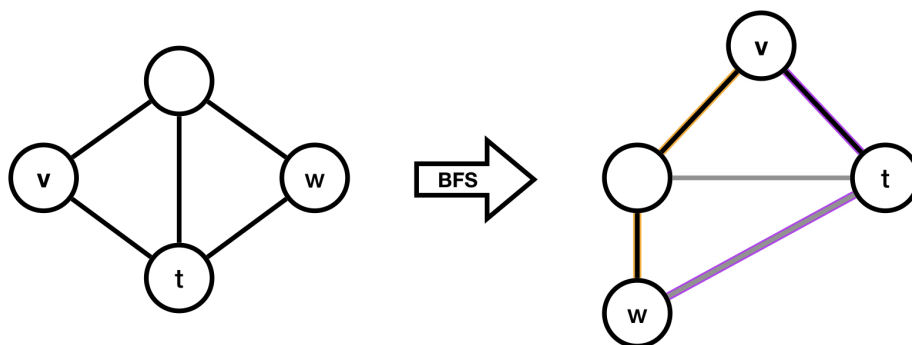


Figura 1: Ejemplo de dos caminos de longitud mínima en un árbol BFS

En la figura 1 vemos un ejemplo de un grafo no geodésico. En el árbol BFS, están en negro las aristas que forman parte de él, y en gris aquellas aristas del grafo que no fueron elegidas por el algoritmo. En naranja y violeta están mostrados los dos caminos de longitud mínima; uno perteneciente al árbol y otro con una arista alternativa.

Teniendo en cuenta esto, para determinar si un grafo es geodésico debemos buscar, para cada vértice  $v$ , aristas  $tw$  que no pertenezcan a su árbol BFS y que cumplan que  $d(v, w) = d(v, t) + 1$ . Si hallamos alguna, el grafo no es geodésico y devolvemos los dos caminos de longitud mínima entre  $v$  y  $w$ .

### Complejidad

El algoritmo propuesto realiza  $n$  veces BFS, que tiene complejidad  $O(n + m)$  representando el grafo como lista de adyacencias. Notemos que el input del problema es un grafo conexo, lo que implica que  $m \geq n - 1$ . De esta forma, la complejidad de ejecutar  $n$  veces BFS queda en  $O(nm)$ :

$$\begin{aligned} m \geq n - 1 &\Rightarrow nm + n \geq n^2 \\ O(n(n + m)) &= O(n^2 + nm) = O(2nm + n) = O(nm) \end{aligned}$$

Luego, debemos chequear una condición (en  $O(1)$ ) sobre todas las aristas, para cada vértice. Si además de la lista de adyacencias guardamos una lista de aristas cuando leemos el grafo, lo podemos hacer en  $O(nm)$ .

Reconstruir los caminos si el grafo no es geodésico es a lo sumo  $O(n)$ , y la ejecución de BFS nos devuelve los caminos de longitud mínima entre todo par de vértices, que es lo que devolvemos en el otro caso.

Finalmente, todo el algoritmo se realiza en tiempo  $O(nm)$ .

## 2.3. Instancias de prueba

Para testear la implementación, generamos 100 grafos geodésicos y 100 no geodésicos. Consideramos para esto los **grafos ciclos**, que son aquellos que forman un único ciclo que pasa exactamente una vez por todos los nodos.

Observemos que, en estos grafos, si la cantidad de nodos es par entonces hay dos caminos de longitud mínima entre todos los pares de vértices:

Sea  $G$  un grafo ciclo con  $n$  par. Tenemos que  $m = n$ ; y si  $C$  es el ciclo formado por el grafo entonces  $l(C) = m$ .

Dado  $v$ , tomamos  $w$  tal que  $l(P_{vw}) = \frac{m}{2}$ . Como el grafo es un ciclo, sabemos que existe otro camino de  $w$  a  $v$ , y además  $l(P_{wv}) = \frac{m}{2}$ .

Como sólo hay dos caminos entre todo par de vértices, entonces ambos son caminos de longitud mínima.

Por otro lado, si la cantidad de nodos es impar entonces el grafo es geodésico.

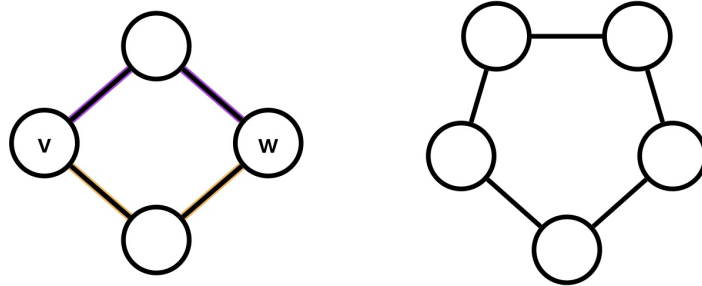


Figura 2: Ejemplo de grafos ciclos

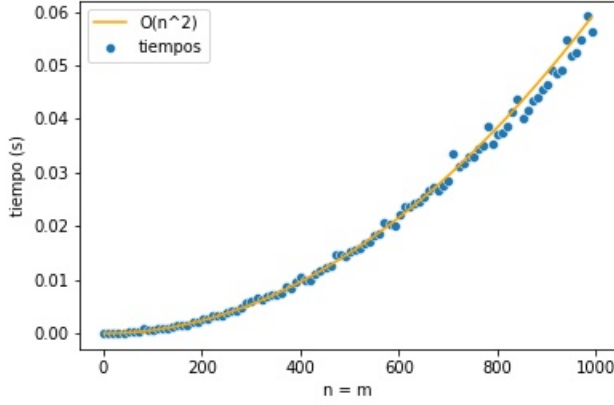
Como en estos casos  $n$  y  $m$  son iguales, generamos además 99 instancias con  $n = 100$  y  $m$  variable: la instancia  $i$  tiene para cada nodo  $j \leq i$  una arista por cada nodo. De esta forma la última instancia es el grafo  $K_{100}$ .

## 2.4. Resultados

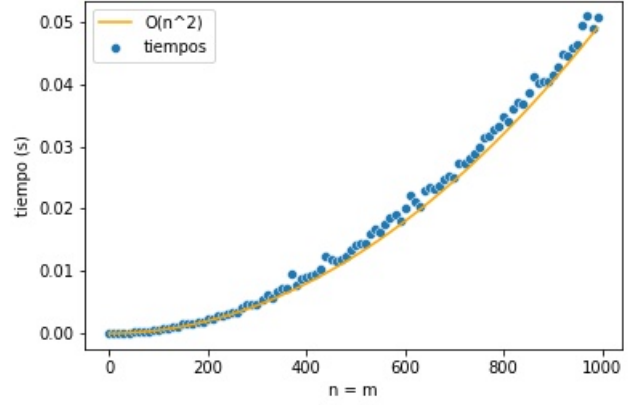
En las tablas de resultados ubicadas en el apéndice se puede ver que el algoritmo clasifica correctamente grafos en geodésicos y no geodésicos.

Como muestran los gráficos, el tiempo de ejecución coincide con la complejidad teórica  $O(nm) = O(n^2)$ .

Se puede ver como, cuando  $n$  esta fijo, el tiempo de ejecución es lineal en  $m$ .



(a) Grafos geodésicos



(b) Grafos no geodésicos

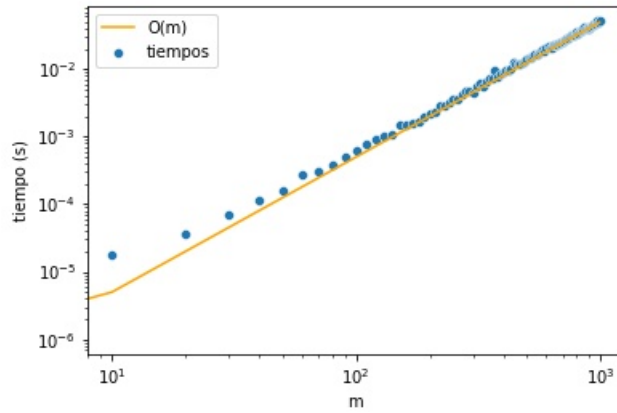


Figura 4: Grafos con 100 nodos y  $m$  aristas

### 3. Ejercicio 2

#### 3.1. Descripción del problema

Dada una matriz  $M \in \{0, 1\}^{m \times n}$ , el grafo  $G(M)$  es aquel que tiene un nodo por cada posición de  $M$  que tiene un 1, y en el que sus nodos son adyacentes si las posiciones de la matriz también lo son.

El objetivo del problema es contar la cantidad de componentes conexas de  $G(M)$ , usando sólo  $O(n)$  espacio.

#### 3.2. El algoritmo

Para resolver el problema usaremos la estructura de **Union-Find**.

Como la restricción de espacio es  $O(n)$ , solo podemos tener una cantidad constante de filas en memoria y UF puede tener información sólo sobre esas filas. Entonces, lo que hacemos es tener dos filas en memoria y un UF de tamaño  $2n$ .

En `filas[0]` y `filas[1]` vamos guardando las dos filas que conocemos; los elementos de 0 a  $n - 1$  de UF se corresponden con las posiciones de `filas[0]`, y los de  $n$  a  $2n - 1$  con las de `filas[1]`.

Al inicio de cada iteración se cumple el siguiente invariante: UF devuelve la misma etiqueta para las

posiciones de `filas[0]` que pertenecen a la misma componente conexa.

Suponiendo que eso se cumple, una iteración realiza los siguientes pasos:

1. lee la siguiente fila de  $M$  y la guarda en `filas[1]`
2. recorre `filas[1]` de a dos posiciones, y si ambas son 1 realiza `unite`
3. recorre `filas[0]` y `filas[1]` a la vez, y si ambas posiciones son 1 realiza `unite`
4. se actualiza el UF para que mantenga la información de las componentes conexas cuando se descarte la fila más vieja
5. se descarta `filas[0]` y se copia `filas[1]` en `filas[0]`

Actualizar el UF implica mantener el invariante que mencionamos previamente:

1. se crea un nuevo UF
2. para cada par de nodos  $i, j$  pertenecientes a la misma componente conexa de `filas[1]` se realiza `unite` en el nuevo UF de los nodos  $i - n$  y  $j - n$ , es decir, de sus posiciones correspondientes en `filas[0]`
3. se reemplaza el UF viejo por el nuevo

Veamos un ejemplo:

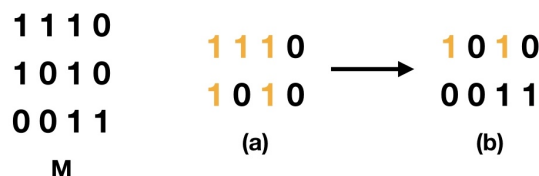


Figura 5: Ejemplo de paso de una iteración a la siguiente

Como podemos ver, las posiciones 0 y 2 de la primer fila en **(b)** pertenecen a la misma componente conexa por más que no lo parezca con la información que hay en memoria en ese momento; y por esta razón es necesario actualizar el UF cada vez que se descarta una fila.

Observemos lo siguiente: dos nodos en  $G(M)$  son adyacentes si y sólo si sus filas en  $M$  son consecutivas. Esto nos permite afirmar que si una componente conexa no tiene nodos en `filas[1]`, entonces no tendrá más nodos de los que ya tiene. Sabiendo esto, cada vez que descartamos una fila podemos contar la cantidad de componentes conexas que ya fueron completadas. De esta forma, luego de haber leído todas las filas obtenemos la cantidad de componentes conexas del grafo.

## Complejidad

Habiéndonos asegurado de cumplir con la cota de complejidad espacial, estudiemos su complejidad temporal.

- leer una fila y hacer `unite` entre sus elementos es  $O(n\alpha^{-1}(n))$
- recorrer ambas filas y hacer `unite` entre ellas es  $O(n\alpha^{-1}(n))$
- contar la cantidad de componentes conexas que se completaron es  $O(n\alpha^{-1}(n))$
- crear un nuevo UF y actualizarlo es  $O(n\alpha^{-1}(n))$
- descartar una fila y copiar la que queda es  $O(n)$

Todas estas operaciones se realizan  $m$  veces, obteniendo una complejidad total de  $O(mn\alpha^{-1}(n))$ , con  $\alpha$  la función de Ackerman.

### 3.3. Instancias de prueba

Para estudiar la correctitud de la implementacion, hicimos algunas instancias de prueba con configuraciones particulares de nodos:

- `sin-nodos` es una matriz de  $5 \times 5$  llena de 0s
- `todos-nodos` es una matriz de  $5 \times 5$  llena de 1s
- `una-columna` es una matriz de  $10 \times 1$  con 1s y 0s sin ningún orden particular
- `una-fila` es una matriz de  $1 \times 10$  con 1s y 0s sin ningún orden particular
- `disconexo` es una matriz de  $5 \times 5$  con 1s y 0s alternados en filas y columnas
- `bordes` es una matriz de  $5 \times 5$  con 1s en los bordes de la matriz y 0s en el medio

Además, generamos 200 instancias aleatorias: 100 con  $m$  fijo y  $n$  creciente y 100 con  $n$  fijo y  $m$  creciente. El valor de cada posición de la matriz se calculó con la función `randint` del módulo `random` de python.

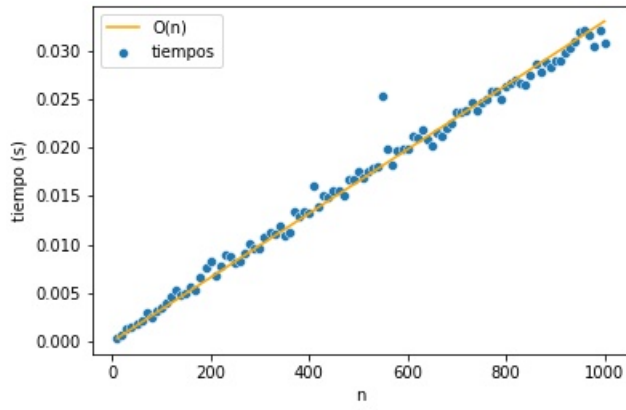
### 3.4. Resultados

Obtuvimos los resultados esperados en los casos particulares.

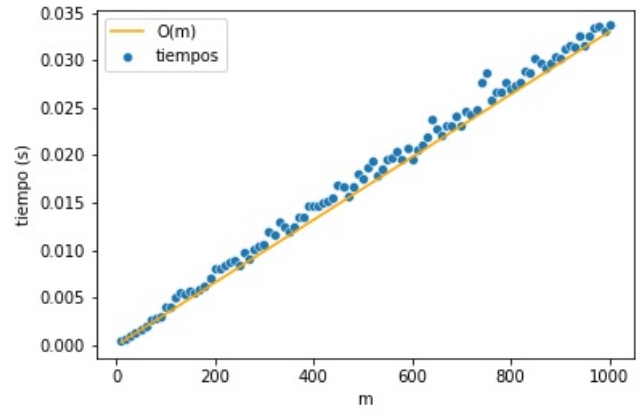
Instancia	n	m	Resultado	Tiempo
bordes	5	5	1	0.00
disconexo	5	5	13	0.00
sin-nodos	5	5	0	0.00
todos-nodos	5	5	1	0.00
una-columna	1	10	3	0.00
una-fila	10	1	3	0.00

Como se puede ver en las figuras 6a y 6b, al dejar una de las variables fijas el tiempo de ejecución es prácticamente lineal en la otra, que es lo esperado según la complejidad teórica  $O(nm\alpha^{-1}(n))$





(a) Cantidad de filas fija,  $m = 50$



(b) Cantidad de columnas fija,  $n = 50$

Figura 6

## 4. Ejercicio 3

### 4.1. Descripción del problema

Dado un digrafo conexo y pesado, queremos hallar el camino mínimo entre todo par de vértices.

### 4.2. El algoritmo

Para resolver este problema utilizaremos el algoritmo de **Johnson**.

Dado un digrafo  $G = (V, E)$  y una función de peso  $w : V \rightarrow \mathbb{Z}$ , definimos  $G' = (V', E')$  de la siguiente forma:

- $V' = V \cup \{s\}$ , con  $s \notin V$
- $E' = E \cup \{(s, v) : v \in V\}$
- $w(s, v) = 0$  para todo  $v \in V$

Ejecutamos **Bellman-Ford** en  $G'$  desde  $s$ , para calcular las distancias de  $s$  a todos los vértices y determinar si el grafo tiene un ciclo negativo (observemos que  $G'$  tiene un ciclo negativo si y solo si  $G$  también, ya que no hay aristas que lleguen a  $s$ ).

Si no hay ciclos negativos, calculamos una nueva función de peso  $\hat{w}$  para  $G'$  que cumple lo siguiente:

$$\hat{w}(u, v) = w(u, v) + \delta(s, u) - \delta(s, v),$$

con  $\delta$  las distancias obtenidas con Bellman-Ford

Se puede probar que la función  $\hat{w}$  cumple que:

1. para todo par de vértices  $u, v \in V$ , un camino de  $u$  a  $v$  es mínimo con la función de peso  $w$  si y solo si lo es con la función de peso  $\hat{w}$
2.  $\hat{w}(u, v) \geq 0$  para todo  $u, v \in V$

De esta forma, para hallar los caminos mínimos entre todo par de vértices, ejecutamos  $n$  veces **Dijkstra** en  $G$  con la función de peso  $\hat{w}$ , una desde cada vértice en  $V$ .

En caso de haber ciclos negativos, sabemos que hasta la iteración  $n - 1$  de **Bellman-Ford** tenemos  $d(v)$  asociado a cada vértice  $v$  tal que  $d(v)$  es menor o igual a la longitud del camino simple más corto entre  $s$  y  $v$ . En ese contexto, si alguna relajación deriva en un decremento de  $d(v_0)$  para algún vértice  $v_0$ , podemos asegurar que  $v_0$  pertenece a un ciclo negativo dado que  $d(v_0)$  resulta menor a la longitud del camino simple más corto  $\delta(s, v_0)$  y que —producto de la relajación— la cadena de predecesores de  $v_0$  reflejará en ese instante el camino a recorrer para certificar el  $d(v_0)$  obtenido.

Así, el algoritmo para encontrar el ciclo negativo detectado (o cualquier otro) consiste en ejecutar una iteración más de **Bellman-Ford** y, al encontrar una arista  $(u, v)$  tal que  $d(u) + w(u, v) < d(v)$  (la arista permite ser relajada), recorrer la cadena de predecesores de  $v$  hasta encontrar un ciclo.

### Complejidad

Armado el digrafo  $G'$  a partir de  $G$  es  $O(n)$ , y corremos Bellman-Ford sobre él en  $O(nm)$ .

Calcular la nueva función de peso es  $O(m)$  y luego corremos Dijkstra, con complejidad  $O(m \log n)$ ,  $n$  veces.

Finalmente, la complejidad temporal total del algoritmo es  $O(nm \log n)$ .

Para la complejidad espacial tenemos la complejidad de guardar el grafo y guardar el ciclo negativo, caso de haberlo.

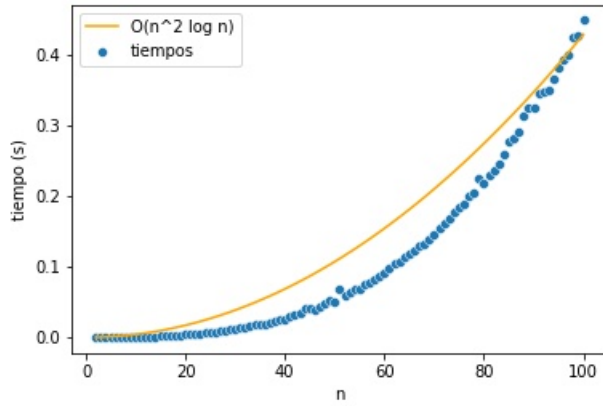
Como el grafo es conexo, entonces  $m \geq n - 1$ . Por lo tanto la complejidad espacial de guardar el mismo es  $O(n + m) = O(m)$ .

La complejidad de guardar el ciclo negativo es  $O(n)$ . Finalmente la complejidad espacial de todo el algoritmo es  $O(m) + O(n) = O(m)$ .

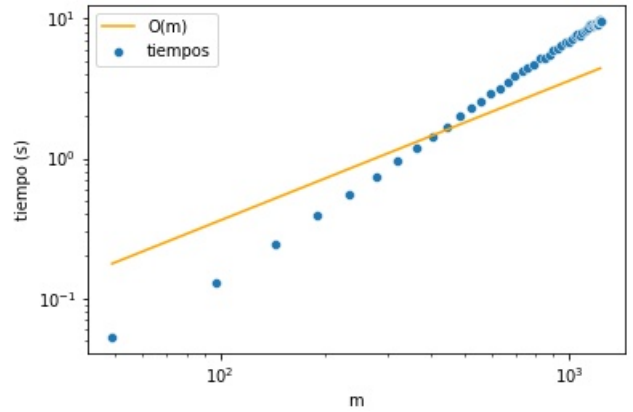
### 4.3. Instancias

Para analizar los tiempos de ejecución del algoritmo, generamos dos sets de instancias: 100 de grafos ciclos con  $n = m$  crecientes, y 50 con una cantidad fija de nodos  $n = 50$  y cantidad de aristas crecientes como las del ejercicio 1.

### 4.4. Resultados



(a) Grafos ciclos,  $n = m$



(b) Cantidad de nodos fija,  $n = 50$

Figura 7

## 5. Ejercicio 4

### 5.1. Descripción del problema

Dada  $\mathcal{I}$  una familia de intervalos en  $\mathbb{R}$ , decimos que  $\mathcal{D} \subset \mathcal{I}$  es un **conjunto dominante total** (CDT) de  $\mathcal{I}$  si para todo  $I \in \mathcal{I}$  existe  $I' \in \mathcal{D}$  tal que  $I \cap I' \neq \emptyset$  y  $I \neq I'$ .

El objetivo del problema es hallar un CDT de cardinalidad mínima.

### 5.2. El algoritmo

Utilizaremos el algoritmo de Bertossi [1] para resolver el problema.

Sea  $\mathcal{I}$  una familia de  $n$  intervalos tales que  $I_i = [s_i, t_i]$  para cada  $i = 1, \dots, n$ . Asumimos que  $s_i < s_j$  si  $i < j$ .

Si hubiera un  $I_k$  tal que  $I_i \subset I_k$  para todo  $i$ , es fácil ver que  $I_k$  junto con cualquier otro intervalo forman un CDT de cardinalidad mínima.

Si no, definimos  $\mathcal{I}' = \mathcal{I} \cup \{I_0, I_{n+1}\}$  donde  $I_0$  es tal que  $t_0 < s_1$  e  $I_{n+1}$  tal que  $t_n < s_{n+1}$ , y definimos un digrafo  $D(N, A)$  de la siguiente manera:

- $i \in N$  si  $I_i \in \mathcal{I}'$  y no está completamente contenido en otro intervalo
- $A = B \cup C$ , con  $B \cap C = \emptyset$
- $ij \in B$  si  $s_i < s_j < t_i < t_j$
- $ij \in C$  si  $t_i < s_j$  y no existe  $I_h$  con  $t_i < s_h < t_h < s_j$
- definimos el peso de las aristas como  $w(ij) = 1$  si  $i \neq 0$  y  $w(0i) = 0$

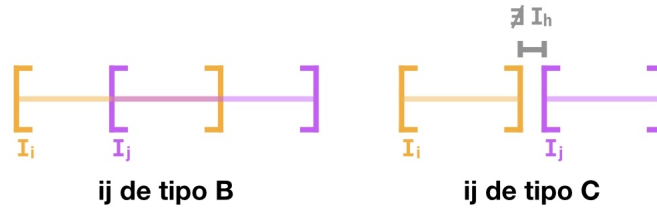


Figura 8: Ejemplo de clasificación de aristas

Notemos que  $D$  no tiene ciclos, ya que  $ij \in A$  si  $s_i < s_j$ , y los intervalos están ordenados crecientemente según  $s_i$ .

Ahora definimos un nuevo digrafo  $D'$  a partir de  $D$ :

- cada nodo  $i \in N \setminus \{0, n+1\}$  se divide en dos nodos  $i_{in}$  e  $i_{out}$
- $i_{in}i_{out} \in A'$  con peso 0
- para cada  $ij \in B$ ,  $i_{out}j_{in} \in A'$  con peso 1
- para cada  $ij \in C$ ,  $i_{in}j_{out} \in A'$  con peso 1

- para cada  $0i \in A$ ,  $0i_{out} \in A'$  con peso 0
- para cada  $i(n+1) \in A$ ,  $i_{in}(n+1) \in A'$  con peso 1

Está probado en el paper de Bertossi que un camino mínimo entre los nodos 0 y  $n+1$  se corresponde con un CDT de cardinalidad mínima, en el cual  $I_i \in \mathcal{I}$  pertenece a él si y sólo si  $i_{in}$ ,  $i_{out}$  o ambos son parte del camino.

## Complejidad

Empezamos ordenando los intervalos según tiempo de inicio, que al estar acotado por  $2n$  lo hacemos con **bucket sort** en  $O(n)$ .

Luego, agregamos  $I_0$  e  $I_{n+1}$  en  $O(1)$  y sacamos los intervalos contenidos en algún otro en  $O(n^2)$ .

El digrafo  $D'$  se puede armar a partir de  $\mathcal{I}$  en  $O(n^2)$ , ya que para cada nodo debemos chequear si corresponde agregar una arista con el resto.

Finalmente, hallamos el camino mínimo desde  $I_0$  a  $I_{n+1}$  en  $O(n^2)$  con el algoritmo de **Dijkstra**, y construimos el CDT en  $O(n)$ .

De esta forma, la complejidad total del algoritmo es  $O(n^2)$ .

## 5.3. Instancias de prueba

Para estudiar el tiempo de ejecución del algoritmo generamos 100 instancias de tamaño creciente. En ellas, todos los intervalos son de la pinta  $[s, s+3]$  con  $s = 1, \dots, n$ . De esta forma cada intervalo se interseque con el siguiente.

Además, los intervalos fueron desordenados con la función **shuffle** del método **random** de python.

## 5.4. Resultados

Como podemos ver en la figura 9, el tiempo de ejecución coincide con la complejidad teórica  $O(n^2)$ .

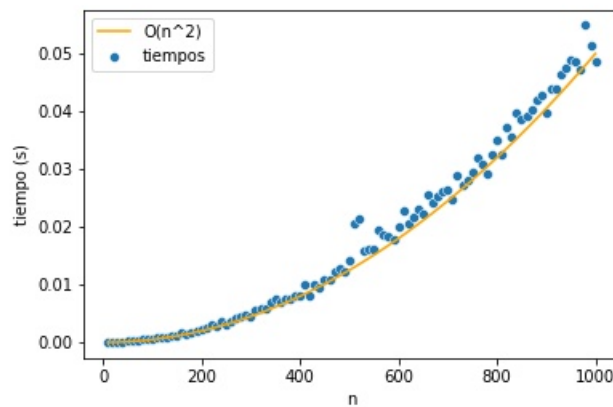


Figura 9: Tiempo de ejecución en función de  $n$

## 6. Conclusión

Los problemas del grafo geodésico y las componentes conexas de un grafo pueden resolverse utilizando herramientas vistas en clase, tales como **BFS** ó **Union Find**.

Por otra parte, las implementaciones de los algoritmos de Johnson y Bertossi resultaron eficientes en la práctica. En particular el algoritmo de Johnson resulta competitivo frente al algoritmo de Floyd-Warshall.

Sobre todos los ejercicios se puede apreciar en los gráficos como la estimación de tiempo basada en la complejidad temporal de los algoritmos resulta un certero predictor de tiempo en la práctica.

## Referencias

- [1] Alan A. Bertossi. Total domination in interval graphs.

## 7. Apéndice

### Ejercicio 1

Instancia	n	m	Resultado	Tiempo
geo-001	1	1	1	0.00
geo-011	11	11	1	0.00
geo-021	21	21	1	0.00
geo-031	31	31	1	0.00
geo-041	41	41	1	0.00
geo-051	51	51	1	0.00
geo-061	61	61	1	0.00
geo-071	71	71	1	0.00
geo-081	81	81	1	0.00
geo-091	91	91	1	0.00
geo-101	101	101	1	0.00
geo-111	111	111	1	0.00
geo-121	121	121	1	0.00
geo-131	131	131	1	0.00
geo-141	141	141	1	0.00
geo-151	151	151	1	0.00
geo-161	161	161	1	0.00
geo-171	171	171	1	0.00
geo-181	181	181	1	0.00
geo-191	191	191	1	0.00
geo-201	201	201	1	0.00
geo-211	211	211	1	0.00
geo-221	221	221	1	0.00
geo-231	231	231	1	0.00
geo-241	241	241	1	0.00
geo-251	251	251	1	0.00
geo-261	261	261	1	0.00
geo-271	271	271	1	0.00
geo-281	281	281	1	0.00
geo-291	291	291	1	0.01
geo-301	301	301	1	0.01
geo-311	311	311	1	0.01
geo-321	321	321	1	0.01
geo-331	331	331	1	0.01
geo-341	341	341	1	0.01
geo-351	351	351	1	0.01
geo-361	361	361	1	0.01
geo-371	371	371	1	0.01
geo-381	381	381	1	0.01
geo-391	391	391	1	0.01
geo-401	401	401	1	0.01
geo-411	411	411	1	0.01
geo-421	421	421	1	0.01
geo-431	431	431	1	0.01
geo-441	441	441	1	0.01

<b>Instancia</b>	<b>n</b>	<b>m</b>	<b>Resultado</b>	<b>Tiempo</b>
geo-451	451	451	1	0.01
geo-461	461	461	1	0.01
geo-471	471	471	1	0.01
geo-481	481	481	1	0.01
geo-491	491	491	1	0.01
geo-501	501	501	1	0.02
geo-511	511	511	1	0.02
geo-521	521	521	1	0.02
geo-531	531	531	1	0.02
geo-541	541	541	1	0.02
geo-551	551	551	1	0.02
geo-561	561	561	1	0.02
geo-571	571	571	1	0.02
geo-581	581	581	1	0.02
geo-591	591	591	1	0.02
geo-601	601	601	1	0.02
geo-611	611	611	1	0.02
geo-621	621	621	1	0.02
geo-631	631	631	1	0.02
geo-641	641	641	1	0.02
geo-651	651	651	1	0.03
geo-661	661	661	1	0.03
geo-671	671	671	1	0.03
geo-681	681	681	1	0.03
geo-691	691	691	1	0.03
geo-701	701	701	1	0.03
geo-711	711	711	1	0.03
geo-721	721	721	1	0.03
geo-731	731	731	1	0.03
geo-741	741	741	1	0.03
geo-751	751	751	1	0.03
geo-761	761	761	1	0.03
geo-771	771	771	1	0.03
geo-781	781	781	1	0.04
geo-791	791	791	1	0.04
geo-801	801	801	1	0.04
geo-811	811	811	1	0.04
geo-821	821	821	1	0.04
geo-831	831	831	1	0.04
geo-841	841	841	1	0.04
geo-851	851	851	1	0.04
geo-861	861	861	1	0.04
geo-871	871	871	1	0.04
geo-881	881	881	1	0.04
geo-891	891	891	1	0.05
geo-901	901	901	1	0.05
geo-911	911	911	1	0.05
geo-921	921	921	1	0.05



<b>Instancia</b>	<b>n</b>	<b>m</b>	<b>Resultado</b>	<b>Tiempo</b>
geo-931	931	931	1	0.05
geo-941	941	941	1	0.05
geo-951	951	951	1	0.05
geo-961	961	961	1	0.05
geo-971	971	971	1	0.05
geo-981	981	981	1	0.06
geo-991	991	991	1	0.06

<b>Instancia</b>	<b>n</b>	<b>m</b>	<b>Resultado</b>	<b>Tiempo</b>
no-geo-010	10	10	0	0.00
no-geo-020	20	20	0	0.00
no-geo-030	30	30	0	0.00
no-geo-040	40	40	0	0.00
no-geo-050	50	50	0	0.00
no-geo-060	60	60	0	0.00
no-geo-070	70	70	0	0.00
no-geo-080	80	80	0	0.00
no-geo-090	90	90	0	0.00
no-geo-100	100	100	0	0.00
no-geo-110	110	110	0	0.00
no-geo-120	120	120	0	0.00
no-geo-130	130	130	0	0.00
no-geo-140	140	140	0	0.00
no-geo-150	150	150	0	0.00
no-geo-160	160	160	0	0.00
no-geo-170	170	170	0	0.00
no-geo-180	180	180	0	0.00
no-geo-190	190	190	0	0.00
no-geo-200	200	200	0	0.00
no-geo-210	210	210	0	0.00
no-geo-220	220	220	0	0.00
no-geo-230	230	230	0	0.00
no-geo-240	240	240	0	0.00
no-geo-250	250	250	0	0.00
no-geo-260	260	260	0	0.00
no-geo-270	270	270	0	0.00
no-geo-280	280	280	0	0.00
no-geo-290	290	290	0	0.00
no-geo-300	300	300	0	0.00
no-geo-310	310	310	0	0.01
no-geo-320	320	320	0	0.01
no-geo-330	330	330	0	0.01
no-geo-340	340	340	0	0.01
no-geo-350	350	350	0	0.01
no-geo-360	360	360	0	0.01
no-geo-370	370	370	0	0.01
no-geo-380	380	380	0	0.01
no-geo-390	390	390	0	0.01

<b>Instancia</b>	<b>n</b>	<b>m</b>	<b>Resultado</b>	<b>Tiempo</b>
no-geo-400	400	400	0	0.01
no-geo-410	410	410	0	0.01
no-geo-420	420	420	0	0.01
no-geo-430	430	430	0	0.01
no-geo-440	440	440	0	0.01
no-geo-450	450	450	0	0.01
no-geo-460	460	460	0	0.01
no-geo-470	470	470	0	0.01
no-geo-480	480	480	0	0.01
no-geo-490	490	490	0	0.01
no-geo-500	500	500	0	0.01
no-geo-510	510	510	0	0.01
no-geo-520	520	520	0	0.01
no-geo-530	530	530	0	0.02
no-geo-540	540	540	0	0.02
no-geo-550	550	550	0	0.02
no-geo-560	560	560	0	0.02
no-geo-570	570	570	0	0.02
no-geo-580	580	580	0	0.02
no-geo-590	590	590	0	0.02
no-geo-600	600	600	0	0.02
no-geo-610	610	610	0	0.02
no-geo-620	620	620	0	0.02
no-geo-630	630	630	0	0.02
no-geo-640	640	640	0	0.02
no-geo-650	650	650	0	0.02
no-geo-660	660	660	0	0.02
no-geo-670	670	670	0	0.02
no-geo-680	680	680	0	0.02
no-geo-690	690	690	0	0.03
no-geo-700	700	700	0	0.03
no-geo-710	710	710	0	0.03
no-geo-720	720	720	0	0.03
no-geo-730	730	730	0	0.03
no-geo-740	740	740	0	0.03
no-geo-750	750	750	0	0.03
no-geo-760	760	760	0	0.03
no-geo-770	770	770	0	0.03
no-geo-780	780	780	0	0.03
no-geo-790	790	790	0	0.03
no-geo-800	800	800	0	0.03
no-geo-810	810	810	0	0.03
no-geo-820	820	820	0	0.04
no-geo-830	830	830	0	0.04
no-geo-840	840	840	0	0.04
no-geo-850	850	850	0	0.04
no-geo-860	860	860	0	0.04
no-geo-870	870	870	0	0.04
no-geo-880	880	880	0	0.04

<b>Instancia</b>	<b>n</b>	<b>m</b>	<b>Resultado</b>	<b>Tiempo</b>
no-geo-890	890	890	0	0.04
no-geo-900	900	900	0	0.04
no-geo-910	910	910	0	0.04
no-geo-920	920	920	0	0.04
no-geo-930	930	930	0	0.04
no-geo-940	940	940	0	0.05
no-geo-950	950	950	0	0.05
no-geo-960	960	960	0	0.05
no-geo-970	970	970	0	0.05
no-geo-980	980	980	0	0.05
no-geo-990	990	990	0	0.05

<b>Instancia</b>	<b>n</b>	<b>m</b>	<b>Resultado</b>	<b>Tiempo</b>
n-fijo-0099	100	99	1	0.00
n-fijo-0197	100	197	0	0.00
n-fijo-0294	100	294	0	0.00
n-fijo-0390	100	390	0	0.00
n-fijo-0485	100	485	0	0.00
n-fijo-0579	100	579	0	0.00
n-fijo-0672	100	672	0	0.00
n-fijo-0764	100	764	0	0.00
n-fijo-0855	100	855	0	0.00
n-fijo-0945	100	945	0	0.00
n-fijo-1034	100	1034	0	0.00
n-fijo-1122	100	1122	0	0.00
n-fijo-1209	100	1209	0	0.01
n-fijo-1295	100	1295	0	0.00
n-fijo-1380	100	1380	0	0.01
n-fijo-1464	100	1464	0	0.00
n-fijo-1547	100	1547	0	0.00
n-fijo-1629	100	1629	0	0.01
n-fijo-1710	100	1710	0	0.00
n-fijo-1790	100	1790	0	0.00
n-fijo-1869	100	1869	0	0.00
n-fijo-1947	100	1947	0	0.00
n-fijo-2024	100	2024	0	0.00
n-fijo-2100	100	2100	0	0.00
n-fijo-2175	100	2175	0	0.00
n-fijo-2249	100	2249	0	0.00
n-fijo-2322	100	2322	0	0.00
n-fijo-2394	100	2394	0	0.00
n-fijo-2465	100	2465	0	0.00
n-fijo-2535	100	2535	0	0.00
n-fijo-2604	100	2604	0	0.00
n-fijo-2672	100	2672	0	0.00
n-fijo-2739	100	2739	0	0.00
n-fijo-2805	100	2805	0	0.00
n-fijo-2870	100	2870	0	0.00

Instancia	n	m	Resultado	Tiempo
n-fijo-2934	100	2934	0	0.00
n-fijo-2997	100	2997	0	0.01
n-fijo-3059	100	3059	0	0.00
n-fijo-3120	100	3120	0	0.01
n-fijo-3180	100	3180	0	0.01
n-fijo-3239	100	3239	0	0.01
n-fijo-3297	100	3297	0	0.01
n-fijo-3354	100	3354	0	0.01
n-fijo-3410	100	3410	0	0.00
n-fijo-3465	100	3465	0	0.00
n-fijo-3519	100	3519	0	0.01
n-fijo-3572	100	3572	0	0.00
n-fijo-3624	100	3624	0	0.00
n-fijo-3675	100	3675	0	0.01
n-fijo-3725	100	3725	0	0.01
n-fijo-3774	100	3774	0	0.01
n-fijo-3822	100	3822	0	0.00
n-fijo-3869	100	3869	0	0.01
n-fijo-3915	100	3915	0	0.01
n-fijo-3960	100	3960	0	0.01
n-fijo-4004	100	4004	0	0.01
n-fijo-4047	100	4047	0	0.01
n-fijo-4089	100	4089	0	0.01
n-fijo-4130	100	4130	0	0.01
n-fijo-4170	100	4170	0	0.01
n-fijo-4209	100	4209	0	0.01
n-fijo-4247	100	4247	0	0.01
n-fijo-4284	100	4284	0	0.01
n-fijo-4320	100	4320	0	0.01
n-fijo-4355	100	4355	0	0.01
n-fijo-4389	100	4389	0	0.01
n-fijo-4422	100	4422	0	0.01
n-fijo-4454	100	4454	0	0.01
n-fijo-4485	100	4485	0	0.01
n-fijo-4515	100	4515	0	0.01
n-fijo-4544	100	4544	0	0.01
n-fijo-4572	100	4572	0	0.01
n-fijo-4599	100	4599	0	0.01
n-fijo-4625	100	4625	0	0.01
n-fijo-4650	100	4650	0	0.01
n-fijo-4674	100	4674	0	0.01
n-fijo-4697	100	4697	0	0.01
n-fijo-4719	100	4719	0	0.01
n-fijo-4740	100	4740	0	0.01
n-fijo-4760	100	4760	0	0.01
n-fijo-4779	100	4779	0	0.01
n-fijo-4797	100	4797	0	0.02
n-fijo-4814	100	4814	0	0.01
n-fijo-4830	100	4830	0	0.01

<b>Instancia</b>	<b>n</b>	<b>m</b>	<b>Resultado</b>	<b>Tiempo</b>
n-fijo-4845	100	4845	0	0.01
n-fijo-4859	100	4859	0	0.01
n-fijo-4872	100	4872	0	0.01
n-fijo-4884	100	4884	0	0.01
n-fijo-4895	100	4895	0	0.01
n-fijo-4905	100	4905	0	0.01
n-fijo-4914	100	4914	0	0.01
n-fijo-4922	100	4922	0	0.01
n-fijo-4929	100	4929	0	0.01
n-fijo-4935	100	4935	0	0.01
n-fijo-4940	100	4940	0	0.01
n-fijo-4944	100	4944	0	0.01
n-fijo-4947	100	4947	0	0.01
n-fijo-4949	100	4949	0	0.01
n-fijo-4950	100	4950	1	0.01

## Ejercicio 2

<b>Instancia</b>	<b>n</b>	<b>m</b>	<b>Resultado</b>	<b>Tiempo</b>
m-fijo-0010	10	50	54	0.00
m-fijo-0020	20	50	72	0.00
m-fijo-0030	30	50	123	0.00
m-fijo-0040	40	50	150	0.00
m-fijo-0050	50	50	211	0.00
m-fijo-0060	60	50	208	0.00
m-fijo-0070	70	50	228	0.00
m-fijo-0080	80	50	261	0.00
m-fijo-0090	90	50	344	0.00
m-fijo-0100	100	50	359	0.00
m-fijo-0110	110	50	394	0.00
m-fijo-0120	120	50	427	0.00
m-fijo-0130	130	50	513	0.01
m-fijo-0140	140	50	483	0.00
m-fijo-0150	150	50	521	0.01
m-fijo-0160	160	50	511	0.01
m-fijo-0170	170	50	565	0.01
m-fijo-0180	180	50	600	0.01
m-fijo-0190	190	50	638	0.01
m-fijo-0200	200	50	681	0.01
m-fijo-0210	210	50	750	0.01
m-fijo-0220	220	50	797	0.01
m-fijo-0230	230	50	775	0.01
m-fijo-0240	240	50	872	0.01
m-fijo-0250	250	50	880	0.01
m-fijo-0260	260	50	960	0.01
m-fijo-0270	270	50	976	0.01
m-fijo-0280	280	50	939	0.01
m-fijo-0290	290	50	989	0.01

<b>Instancia</b>	<b>n</b>	<b>m</b>	<b>Resultado</b>	<b>Tiempo</b>
m-fijo-0300	300	50	1058	0.01
m-fijo-0310	310	50	1011	0.01
m-fijo-0320	320	50	1125	0.01
m-fijo-0330	330	50	1140	0.01
m-fijo-0340	340	50	1164	0.01
m-fijo-0350	350	50	1190	0.01
m-fijo-0360	360	50	1242	0.01
m-fijo-0370	370	50	1275	0.01
m-fijo-0380	380	50	1313	0.01
m-fijo-0390	390	50	1337	0.01
m-fijo-0400	400	50	1475	0.01
m-fijo-0410	410	50	1473	0.02
m-fijo-0420	420	50	1416	0.01
m-fijo-0430	430	50	1523	0.02
m-fijo-0440	440	50	1529	0.01
m-fijo-0450	450	50	1618	0.02
m-fijo-0460	460	50	1554	0.02
m-fijo-0470	470	50	1609	0.02
m-fijo-0480	480	50	1685	0.02
m-fijo-0490	490	50	1724	0.02
m-fijo-0500	500	50	1787	0.02
m-fijo-0510	510	50	1786	0.02
m-fijo-0520	520	50	1766	0.02
m-fijo-0530	530	50	1799	0.02
m-fijo-0540	540	50	1798	0.02
m-fijo-0550	550	50	1849	0.03
m-fijo-0560	560	50	1953	0.02
m-fijo-0570	570	50	2050	0.02
m-fijo-0580	580	50	1962	0.02
m-fijo-0590	590	50	2088	0.02
m-fijo-0600	600	50	2108	0.02
m-fijo-0610	610	50	2032	0.02
m-fijo-0620	620	50	2119	0.02
m-fijo-0630	630	50	2186	0.02
m-fijo-0640	640	50	2231	0.02
m-fijo-0650	650	50	2247	0.02
m-fijo-0660	660	50	2286	0.02
m-fijo-0670	670	50	2303	0.02
m-fijo-0680	680	50	2403	0.02
m-fijo-0690	690	50	2361	0.02
m-fijo-0700	700	50	2484	0.02
m-fijo-0710	710	50	2544	0.02
m-fijo-0720	720	50	2443	0.02
m-fijo-0730	730	50	2535	0.02
m-fijo-0740	740	50	2672	0.02
m-fijo-0750	750	50	2607	0.02
m-fijo-0760	760	50	2647	0.02
m-fijo-0770	770	50	2710	0.03
m-fijo-0780	780	50	2739	0.03

<b>Instancia</b>	<b>n</b>	<b>m</b>	<b>Resultado</b>	<b>Tiempo</b>
m-fijo-0790	790	50	2683	0.02
m-fijo-0800	800	50	2777	0.03
m-fijo-0810	810	50	2887	0.03
m-fijo-0820	820	50	2905	0.03
m-fijo-0830	830	50	2854	0.03
m-fijo-0840	840	50	3020	0.03
m-fijo-0850	850	50	2900	0.03
m-fijo-0860	860	50	3013	0.03
m-fijo-0870	870	50	2981	0.03
m-fijo-0880	880	50	3102	0.03
m-fijo-0890	890	50	3038	0.03
m-fijo-0900	900	50	3125	0.03
m-fijo-0910	910	50	3119	0.03
m-fijo-0920	920	50	3174	0.03
m-fijo-0930	930	50	3233	0.03
m-fijo-0940	940	50	3139	0.03
m-fijo-0950	950	50	3301	0.03
m-fijo-0960	960	50	3319	0.03
m-fijo-0970	970	50	3389	0.03
m-fijo-0980	980	50	3431	0.03
m-fijo-0990	990	50	3434	0.03
m-fijo-1000	1000	50	3376	0.03

<b>Instancia</b>	<b>n</b>	<b>m</b>	<b>Resultado</b>	<b>Tiempo</b>
n-fijo-0010	50	10	52	0.00
n-fijo-0020	50	20	79	0.00
n-fijo-0030	50	30	129	0.00
n-fijo-0040	50	40	152	0.00
n-fijo-0050	50	50	184	0.00
n-fijo-0060	50	60	230	0.00
n-fijo-0070	50	70	241	0.00
n-fijo-0080	50	80	323	0.00
n-fijo-0090	50	90	344	0.00
n-fijo-0100	50	100	342	0.00
n-fijo-0110	50	110	366	0.00
n-fijo-0120	50	120	416	0.01
n-fijo-0130	50	130	480	0.01
n-fijo-0140	50	140	503	0.01
n-fijo-0150	50	150	517	0.01
n-fijo-0160	50	160	538	0.01
n-fijo-0170	50	170	617	0.01
n-fijo-0180	50	180	648	0.01
n-fijo-0190	50	190	661	0.01
n-fijo-0200	50	200	686	0.01
n-fijo-0210	50	210	703	0.01
n-fijo-0220	50	220	734	0.01
n-fijo-0230	50	230	859	0.01
n-fijo-0240	50	240	787	0.01

<b>Instancia</b>	<b>n</b>	<b>m</b>	<b>Resultado</b>	<b>Tiempo</b>
n-fijo-0250	50	250	864	0.01
n-fijo-0260	50	260	906	0.01
n-fijo-0270	50	270	917	0.01
n-fijo-0280	50	280	980	0.01
n-fijo-0290	50	290	1042	0.01
n-fijo-0300	50	300	1056	0.01
n-fijo-0310	50	310	1160	0.01
n-fijo-0320	50	320	1163	0.01
n-fijo-0330	50	330	1127	0.01
n-fijo-0340	50	340	1139	0.01
n-fijo-0350	50	350	1154	0.01
n-fijo-0360	50	360	1266	0.01
n-fijo-0370	50	370	1234	0.01
n-fijo-0380	50	380	1399	0.01
n-fijo-0390	50	390	1307	0.01
n-fijo-0400	50	400	1404	0.01
n-fijo-0410	50	410	1526	0.01
n-fijo-0420	50	420	1502	0.02
n-fijo-0430	50	430	1481	0.02
n-fijo-0440	50	440	1508	0.02
n-fijo-0450	50	450	1610	0.02
n-fijo-0460	50	460	1692	0.02
n-fijo-0470	50	470	1635	0.02
n-fijo-0480	50	480	1633	0.02
n-fijo-0490	50	490	1739	0.02
n-fijo-0500	50	500	1839	0.02
n-fijo-0510	50	510	1783	0.02
n-fijo-0520	50	520	1783	0.02
n-fijo-0530	50	530	1862	0.02
n-fijo-0540	50	540	1863	0.02
n-fijo-0550	50	550	1944	0.02
n-fijo-0560	50	560	1977	0.02
n-fijo-0570	50	570	2118	0.02
n-fijo-0580	50	580	2060	0.02
n-fijo-0590	50	590	2015	0.02
n-fijo-0600	50	600	2123	0.02
n-fijo-0610	50	610	2184	0.02
n-fijo-0620	50	620	2068	0.02
n-fijo-0630	50	630	2201	0.02
n-fijo-0640	50	640	2232	0.02
n-fijo-0650	50	650	2305	0.02
n-fijo-0660	50	660	2276	0.02
n-fijo-0670	50	670	2261	0.02
n-fijo-0680	50	680	2402	0.02
n-fijo-0690	50	690	2316	0.02
n-fijo-0700	50	700	2319	0.02
n-fijo-0710	50	710	2417	0.02
n-fijo-0720	50	720	2470	0.02
n-fijo-0730	50	730	2573	0.02



<b>Instancia</b>	<b>n</b>	<b>m</b>	<b>Resultado</b>	<b>Tiempo</b>
n-fijo-0740	50	740	2477	0.03
n-fijo-0750	50	750	2671	0.03
n-fijo-0760	50	760	2655	0.03
n-fijo-0770	50	770	2701	0.03
n-fijo-0780	50	780	2753	0.03
n-fijo-0790	50	790	2688	0.03
n-fijo-0800	50	800	2741	0.03
n-fijo-0810	50	810	2780	0.03
n-fijo-0820	50	820	2963	0.03
n-fijo-0830	50	830	2831	0.03
n-fijo-0840	50	840	2898	0.03
n-fijo-0850	50	850	2880	0.03
n-fijo-0860	50	860	2960	0.03
n-fijo-0870	50	870	3015	0.03
n-fijo-0880	50	880	3016	0.03
n-fijo-0890	50	890	3030	0.03
n-fijo-0900	50	900	3152	0.03
n-fijo-0910	50	910	3171	0.03
n-fijo-0920	50	920	3206	0.03
n-fijo-0930	50	930	3257	0.03
n-fijo-0940	50	940	3212	0.03
n-fijo-0950	50	950	3373	0.03
n-fijo-0960	50	960	3332	0.03
n-fijo-0970	50	970	3318	0.03
n-fijo-0980	50	980	3405	0.03
n-fijo-0990	50	990	3347	0.03
n-fijo-1000	50	1000	3361	0.03

### Ejercicio 3

<b>Instancia</b>	<b>n</b>	<b>m</b>	<b>Resultado</b>	<b>Tiempo</b>
ciclo-003	3	3	1	0.00
ciclo-004	4	4	1	0.00
ciclo-005	5	5	1	0.00
ciclo-006	6	6	1	0.00
ciclo-007	7	7	1	0.00
ciclo-008	8	8	1	0.00
ciclo-009	9	9	1	0.00
ciclo-010	10	10	1	0.00
ciclo-011	11	11	1	0.00
ciclo-012	12	12	1	0.00
ciclo-013	13	13	1	0.00
ciclo-014	14	14	1	0.00
ciclo-015	15	15	1	0.00
ciclo-016	16	16	1	0.00
ciclo-017	17	17	1	0.00
ciclo-018	18	18	1	0.00
ciclo-019	19	19	1	0.00
ciclo-020	20	20	1	0.00
ciclo-021	21	21	1	0.00
ciclo-022	22	22	1	0.00

<b>Instancia</b>	<b>n</b>	<b>m</b>	<b>Resultado</b>	<b>Tiempo</b>
ciclo-024	24	24	1	0.01
ciclo-025	25	25	1	0.01
ciclo-026	26	26	1	0.01
ciclo-027	27	27	1	0.01
ciclo-028	28	28	1	0.01
ciclo-029	29	29	1	0.01
ciclo-030	30	30	1	0.01
ciclo-031	31	31	1	0.01
ciclo-032	32	32	1	0.01
ciclo-033	33	33	1	0.02
ciclo-034	34	34	1	0.02
ciclo-035	35	35	1	0.02
ciclo-036	36	36	1	0.02
ciclo-037	37	37	1	0.02
ciclo-038	38	38	1	0.02
ciclo-039	39	39	1	0.02
ciclo-040	40	40	1	0.03
ciclo-041	41	41	1	0.03
ciclo-042	42	42	1	0.03
ciclo-043	43	43	1	0.04
ciclo-044	44	44	1	0.04
ciclo-045	45	45	1	0.04
ciclo-046	46	46	1	0.04
ciclo-047	47	47	1	0.04
ciclo-048	48	48	1	0.05
ciclo-049	49	49	1	0.05
ciclo-050	50	50	1	0.05
ciclo-051	51	51	1	0.07
ciclo-052	52	52	1	0.06
ciclo-053	53	53	1	0.06
ciclo-054	54	54	1	0.07
ciclo-055	55	55	1	0.07
ciclo-056	56	56	1	0.07
ciclo-057	57	57	1	0.08
ciclo-058	58	58	1	0.08
ciclo-059	59	59	1	0.09
ciclo-060	60	60	1	0.09
ciclo-061	61	61	1	0.10
ciclo-062	62	62	1	0.10
ciclo-063	63	63	1	0.10
ciclo-064	64	64	1	0.11
ciclo-065	65	65	1	0.12
ciclo-066	66	66	1	0.12
ciclo-067	67	67	1	0.13
ciclo-068	68	68	1	0.13
ciclo-069	69	69	1	0.14
ciclo-070	70	70	1	0.15
ciclo-071	71	71	1	0.16
ciclo-072	72	72	1	0.16
ciclo-073	73	73	1	0.17
ciclo-074	74	74	1	0.18
ciclo-075	75	75	1	0.18

<b>Instancia</b>	<b>n</b>	<b>m</b>	<b>Resultado</b>	<b>Tiempo</b>
ciclo-077	77	77	1	0.20
ciclo-078	78	78	1	0.21
ciclo-079	79	79	1	0.23
ciclo-080	80	80	1	0.22
ciclo-081	81	81	1	0.23
ciclo-082	82	82	1	0.24
ciclo-083	83	83	1	0.25
ciclo-084	84	84	1	0.26
ciclo-085	85	85	1	0.28
ciclo-086	86	86	1	0.28
ciclo-087	87	87	1	0.29
ciclo-088	88	88	1	0.31
ciclo-089	89	89	1	0.33
ciclo-090	90	90	1	0.33
ciclo-091	91	91	1	0.35
ciclo-092	92	92	1	0.35
ciclo-093	93	93	1	0.35
ciclo-094	94	94	1	0.37
ciclo-095	95	95	1	0.38
ciclo-096	96	96	1	0.39
ciclo-097	97	97	1	0.40
ciclo-098	98	98	1	0.43
ciclo-099	99	99	1	0.43
ciclo-100	100	100	1	0.45

<b>Instancia</b>	<b>n</b>	<b>m</b>	<b>Resultado</b>	<b>Tiempo</b>
n-fijo-0097	50	97	1	0.13
n-fijo-0144	50	144	1	0.24
n-fijo-0190	50	190	1	0.39
n-fijo-0235	50	235	1	0.55
n-fijo-0279	50	279	1	0.74
n-fijo-0322	50	322	1	0.96
n-fijo-0364	50	364	1	1.20
n-fijo-0405	50	405	1	1.42
n-fijo-0445	50	445	1	1.67
n-fijo-0484	50	484	1	2.03
n-fijo-0522	50	522	1	2.30
n-fijo-0559	50	559	1	2.56
n-fijo-0595	50	595	1	2.89
n-fijo-0630	50	630	1	3.18
n-fijo-0664	50	664	1	3.54
n-fijo-0697	50	697	1	3.87
n-fijo-0729	50	729	1	4.20
n-fijo-0760	50	760	1	4.45
n-fijo-0790	50	790	1	4.75
n-fijo-0819	50	819	1	5.20
n-fijo-0847	50	847	1	5.24
n-fijo-0874	50	874	1	5.55
n-fijo-0900	50	900	1	5.92
n-fijo-0925	50	925	1	6.13
n-fijo-0949	50	949	1	6.45

<b>Instancia</b>	<b>n</b>	<b>m</b>	<b>Resultado</b>	<b>Tiempo</b>
n-fijo-0994	50	994	1	6.88
n-fijo-1015	50	1015	1	7.12
n-fijo-1035	50	1035	1	7.59
n-fijo-1054	50	1054	1	7.70
n-fijo-1072	50	1072	1	7.62
n-fijo-1089	50	1089	1	8.06
n-fijo-1105	50	1105	1	8.11
n-fijo-1120	50	1120	1	8.42
n-fijo-1134	50	1134	1	8.52
n-fijo-1147	50	1147	1	8.98
n-fijo-1159	50	1159	1	9.01
n-fijo-1170	50	1170	1	8.96
n-fijo-1180	50	1180	1	9.12
n-fijo-1189	50	1189	1	9.20
n-fijo-1197	50	1197	1	9.38
n-fijo-1204	50	1204	1	9.19
n-fijo-1210	50	1210	1	9.20
n-fijo-1215	50	1215	1	9.57
n-fijo-1219	50	1219	1	9.72
n-fijo-1222	50	1222	1	9.62
n-fijo-1224	50	1224	1	9.54
n-fijo-1225	50	1225	1	9.49

## Ejercicio 4

<b>Instancia</b>	<b>n</b>	<b>Resultado</b>	<b>Tiempo</b>
cadena-0010	10	3	0.00
cadena-0020	20	5	0.00
cadena-0030	30	8	0.00
cadena-0040	40	10	0.00
cadena-0050	50	12	0.00
cadena-0060	60	14	0.00
cadena-0070	70	16	0.00
cadena-0080	80	18	0.00
cadena-0090	90	20	0.00
cadena-0100	100	23	0.00
cadena-0110	110	25	0.00
cadena-0120	120	28	0.00
cadena-0130	130	30	0.00
cadena-0140	140	32	0.00
cadena-0150	150	34	0.00
cadena-0160	160	36	0.00
cadena-0170	170	38	0.00

<b>Instancia</b>	<b>n</b>	<b>Resultado</b>	<b>Tiempo</b>
cadena-0180	180	40	0.00
cadena-0190	190	43	0.00
cadena-0200	200	45	0.00
cadena-0210	210	48	0.00
cadena-0220	220	50	0.00
cadena-0230	230	52	0.00
cadena-0240	240	54	0.00
cadena-0250	250	56	0.00
cadena-0260	260	58	0.00
cadena-0270	270	60	0.00
cadena-0280	280	63	0.00
cadena-0290	290	65	0.00
cadena-0300	300	68	0.00
cadena-0310	310	70	0.01
cadena-0320	320	72	0.01
cadena-0330	330	74	0.01
cadena-0340	340	76	0.01
cadena-0350	350	78	0.01
cadena-0360	360	80	0.01
cadena-0370	370	83	0.01
cadena-0380	380	85	0.01
cadena-0390	390	88	0.01
cadena-0400	400	90	0.01
cadena-0410	410	92	0.01
cadena-0420	420	94	0.01
cadena-0430	430	96	0.01
cadena-0440	440	98	0.01
cadena-0450	450	100	0.01
cadena-0460	460	103	0.01
cadena-0470	470	105	0.01
cadena-0480	480	108	0.01
cadena-0490	490	110	0.01
cadena-0500	500	112	0.01
cadena-0510	510	114	0.02
cadena-0520	520	116	0.02
cadena-0530	530	118	0.02
cadena-0540	540	120	0.02
cadena-0550	550	123	0.02
cadena-0560	560	125	0.02
cadena-0570	570	128	0.02
cadena-0580	580	130	0.02
cadena-0590	590	132	0.02
cadena-0600	600	134	0.02
cadena-0610	610	136	0.02
cadena-0620	620	138	0.02
cadena-0630	630	140	0.02
cadena-0640	640	143	0.02
cadena-0650	650	145	0.02
cadena-0660	660	148	0.02

<b>Instancia</b>	<b>n</b>	<b>Resultado</b>	<b>Tiempo</b>
cadena-0670	670	150	0.02
cadena-0680	680	152	0.03
cadena-0690	690	154	0.03
cadena-0700	700	156	0.03
cadena-0710	710	158	0.02
cadena-0720	720	160	0.03
cadena-0730	730	163	0.03
cadena-0740	740	165	0.03
cadena-0750	750	168	0.03
cadena-0760	760	170	0.03
cadena-0770	770	172	0.03
cadena-0780	780	174	0.03
cadena-0790	790	176	0.03
cadena-0800	800	178	0.03
cadena-0810	810	180	0.03
cadena-0820	820	183	0.04
cadena-0830	830	185	0.04
cadena-0840	840	188	0.04
cadena-0850	850	190	0.04
cadena-0860	860	192	0.04
cadena-0870	870	194	0.04
cadena-0880	880	196	0.04
cadena-0890	890	198	0.04
cadena-0900	900	200	0.04
cadena-0910	910	203	0.04
cadena-0920	920	205	0.04
cadena-0930	930	208	0.05
cadena-0940	940	210	0.05
cadena-0950	950	212	0.05
cadena-0960	960	214	0.05
cadena-0970	970	216	0.05
cadena-0980	980	218	0.05
cadena-0990	990	220	0.05
cadena-1000	1000	223	0.05