PROGRAMACIÓN DINÁMICA

Aránzazu Jurío

ALGORITMIA

2018/2019

Índice

- Introducción
- Problemas
  - Sucesión de Fibonacci
  - Coeficientes binomiales
  - Subsecuencia común máxima
  - Mochila 0-1
  - El problema del cambio
  - Transformación de cadenas
  - Caminos mínimos entre pares de vértices cualesquiera
  - Multiplicación de matrices

Índice

- Introducción
- Problemas
  - Sucesión de Fibonacci
  - Coeficientes binomiales
  - Subsecuencia común máxima
  - Mochila 0-1
  - El problema del cambio
  - Transformación de cadenas
  - Caminos mínimos entre pares de vértices cualesquiera
  - Multiplicación de matrices

Introducción

- Divide y vencerás: dividir el problema en subproblemas más pequeños independientes de la misma naturaleza, y resolverlos recursivamente
- Si los problemas no son independientes, la solución recursiva no es eficiente, porque repite cálculos
- En estos casos utilizamos **programación dinámica**
- Es eficiente porque guarda las soluciones de los subproblemas en un tabla, para poder reutilizarlos

Introducción

- Muy útil para resolver problemas de optimización: existen varias soluciones, cada una con un valor, y se trata de encontrar la solución con valor óptimo (mínimo o máximo)

- Principio de optimalidad (Bellman, 1957)

  En una secuencia de decisiones óptima toda subsecuencia ha de ser también óptima

- Para poder usar programación dinámica en un problema:
  - La solución es un secuencia de decisiones, una en cada etapa
  - La secuencia de decisiones debe cumplir el principio de optimalidad

Introducción

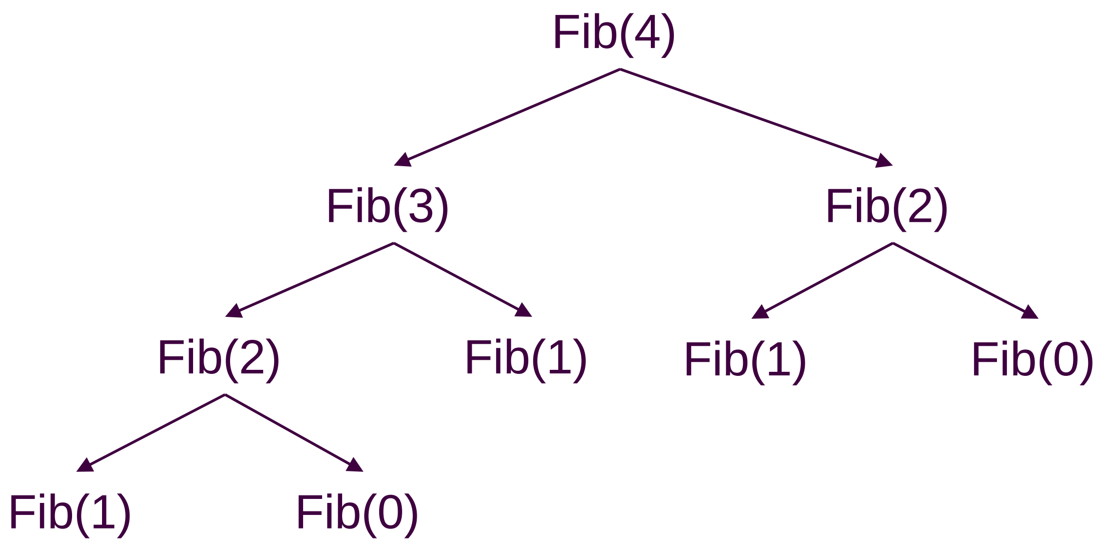
- Pasos para hallar la solución
  - Planteamiento de la solución como sucesión de soluciones, y verificación de que se cumple el principio de optimalidad
  - Definición recursiva de la solución
  - Cálculo del valor de la solución óptima mediante una tabla en sonde se almacenan soluciones a problemas parciales para reutilizar cálculos
  - Construcción de la solución óptima haciendo uso de la información contenida en la tabla anterior

Índice

- Introducción
- Problemas
  - Sucesión de Fibonacci
  - Coeficientes binomiales
  - Subsecuencia común máxima
  - Mochila 0-1
  - El problema del cambio
  - Transformación de cadenas
  - Caminos mínimos entre pares de vértices cualesquiera
  - Multiplicación de matrices

Sucesión de Fibonacci

- Orden exponencial
- $Fib(n) = \begin{cases} 1 & si\ n \in \{0,1\} \\ Fib(n-1) + Fib(n-2) & si\ n > 1 \end{cases}$



Sucesión de Fibonacci

- ¿Cómo podemos hacerlo en tiempo lineal?
- $Fib(n) = \begin{cases} 1 & si\ n \in \{0,1\} \\ Fib(n-1) + Fib(n-2) & si\ n > 1 \end{cases}$

Sucesión de Fibonacci

- ¿Cómo podemos hacerlo en tiempo lineal?
- $Fib(n) = \begin{cases} 1 & si\ n \in \{0,1\} \\ Fib(n-1) + Fib(n-2) & si\ n > 1 \end{cases}$

Fib(0)	Fib(1)	Fib(2)	Fib(3)	…	Fib(n)

Sucesión de Fibonacci

- ¿Podemos mejorar la complejidad espacial?

Sucesión de Fibonacci

- ¿Podemos mejorar la complejidad espacial?
  - De la tabla creada sólo utilizamos los dos últimos elementos
  - Almacenar sólo estos dos elementos

Índice

- Introducción
- Problemas
  - Sucesión de Fibonacci
  - Coeficientes binomiales
  - Subsecuencia común máxima
  - Mochila 0-1
  - El problema del cambio
  - Transformación de cadenas
  - Caminos mínimos entre pares de vértices cualesquiera
  - Multiplicación de matrices

Coeficientes binomiales

- Queremos calcular un coeficiente binomial:
- $$\begin{cases} \binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k} & si\ 0 < k < n \\ 1 & si\ k = 0\ ó\ k = 1 \end{cases}$$
- Implementación recursiva → complejidad exponencial
- Programación dinámica → $O(nk)$
  - ¿Qué valores debemos almacenar?
  - ¿En qué estructura los almacenamos?

Coeficientes binomiales

- Nos inspiramos en el triángulo de Pascal

$$\binom{0}{0}$$
$$\binom{1}{0} \quad \binom{1}{1}$$
$$\binom{2}{0} \quad \binom{2}{1} \quad \binom{2}{2}$$
$$\binom{3}{0} \quad \binom{3}{1} \quad \binom{3}{2} \quad \binom{3}{3}$$
$$\binom{4}{0} \quad \binom{4}{1} \quad \binom{4}{2} \quad \binom{4}{3} \quad \binom{4}{4}$$

				1				
			1		1			
		1		2		1		
	1		3		3		1	
1		4		6		4		1

- Utilizamos una tabla de dos dimensiones para almacenar los resultados parciales

Coeficientes binomiales

- $\binom{n}{0} = 1$ para todo n

	0	1	2	3	4	…	k-1	k
0	1							
1	1							
2	1							
3	1							
4	1							
…	…							
n-1	1							
n	1							

Coeficientes binomiales

- $\binom{n}{n} = 1$ para todo n

	0	1	2	3	4	…	k-1	k
0	1							
1	1	1						
2	1		1					
3	1			1				
4	1				1			
…	…							
n-1	1							
n	1							

Coeficientes binomiales

- $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$
- C[i,j]=C[i-1,j-1]+C[i-1,j]

	0	1	2	3	4	…	k-1	k
0	1							
1	1	1						
2	1	**2**	1					
3	1			1				
4	1				1			
…	…							
n-1	1							
n	1							

Coeficientes binomiales

- $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$
- C[i,j]=C[i-1,j-1]+C[i-1,j]

	0	1	2	3	4	…	k-1	k
0	1							
1	1	1						
2	1	2	1					
3	1	**3**		1				
4	1				1			
…	…							
n-1	1							
n	1							

Coeficientes binomiales

- $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$
- C[i,j]=C[i-1,j-1]+C[i-1,j]

	0	1	2	3	4	…	k-1	k
0	1							
1	1	1						
2	1	2	1					
3	1	3	**3**	1				
4	1				1			
…	…							
n-1	1							
n	1							

Coeficientes binomiales

- $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$
- C[i,j]=C[i-1,j-1]+C[i-1,j]

	0	1	2	3	4	…	k-1	k
0	1							
1	1	1						
2	1	2	1					
3	1	3	3	1				
4	1				1			
…	…							
n-1	1							
n	1							

SOLUCIÓN

Índice

- Introducción
- Problemas
  - Sucesión de Fibonacci
  - Coeficientes binomiales
  - Subsecuencia común máxima
  - Mochila 0-1
  - El problema del cambio
  - Transformación de cadenas
  - Caminos mínimos entre pares de vértices cualesquiera
  - Multiplicación de matrices

Subsecuencia común máxima

- Dada una secuencia $X = \{x_1\ x_2\ \ldots x_m\}$, decimos que $Z = \{z_1\ z_2\ \ldots z_k\}$ es una subsecuencia de X (siendo $k \leq m$) si existe una secuencia creciente $\{i_1\ i_2\ \ldots\ i_k\}$ de índices de X tales que para todo $j = 1, 2, \ldots, k$ tenemos $x_{i_j} = z_j$.

- Dada la secuencia $X = ABCIEDJSAHBCA$
  - $Z_1 = ABCI$ es una subsecuencia de X
  - $Z_2 = CJSH$ es una subsecuencia de X
  - $Z_3 = AA$ es una subsecuencia de X
  - $Z_4 = ABCAS$ **no** es una subsecuencia de X

Subsecuencia común máxima

- Dadas dos secuencias X e Y, decimos que Z es una subsecuencia común de X e Y si es subsecuencia de X y subsecuencia de Y. Deseamos determinar la subsecuencia de longitud máxima común a dos secuencias

- Necesitamos conocer la longitud de la subsecuencia común más larga
- Necesitamos conocer los elementos de dicha subsecuencia

Subsecuencia común máxima

- X = (1 0 0 1 0 1 0 1)
- Y = (0 1 0 1 1 0 1 1 0)

- ¿Cuál es la subsecuencia común máxima?
- ¿Es única?

Subsecuencia común máxima

- X = (1 0 0 1 0 1 0 1)
- Y = (0 1 0 1 1 0 1 1 0)

- ¿Cuál es la subsecuencia común máxima?
- ¿Es única?
  - 1 0 0 1 1 0
  - 0 1 0 1 0 1

Subsecuencia común máxima

- Comenzamos calculando la longitud de la subsecuencia común más larga

- Resuelvo en base a problemas del mismo tipo de menor dimensión, cuya solución he almacenado. ¿Qué tipo de estructura necesito para almacenar estas soluciones?

Subsecuencia común máxima

- Comenzamos calculando la longitud de la subsecuencia común más larga

- Resuelvo en base a problemas del mismo tipo de menor dimensión, cuya solución he almacenado. ¿Qué tipo de estructura necesito para almacenar estas soluciones?
- Casos base
  - Si una de las secuencias tiene longitud 0, entonces la longitud de la mayor subsecuencia común será 0
- ¿Resto de casos?

Subsecuencia común máxima

		0	1	2	3	4	5	6	7	8
			1	0	0	1	0	1	0	1
0		0	0	0	0	0	0	0	0	0
1	0	0								
2	1	0								
3	0	0								
4	1	0								
5	1	0								
6	0	0								
7	1	0								
8	1	0								
9	0	0								

Subsecuencia común máxima

- Para cada subproblema $\boldsymbol{SCM(X_{1..i}, Y_{1..j})}$, estudio el último elemento de cada secuencia
  - $X = (x_1, x_2, \dots, x_{i-1}, \boldsymbol{x_i})$
  - $Y = (y_1, y_2, \dots, y_{j-1}, \boldsymbol{y_j})$

  - Si estos elementos son iguales $\boldsymbol{x_i = y_j}$ entonces la longitud de la secuencia será 1 + más longitud de la subsecuencia común más larga sin tener en cuenta esos elementos
  - $\boldsymbol{SCM\left(X_{1..i}, Y_{1..j}\right) = 1 + SCM(X_{1..i-1}, Y_{1..j-1})}$
  - Si los últimos elementos son distintos $\boldsymbol{x_i \neq y_j}$ entonces la longitud de la secuencia será el máximo entre la solución de toda la cadena X con la cadena Y menos el último elemento, y la solución de la cadena X menos el último elemento con toda la cadena Y
  - $\boldsymbol{SCM\left(X_{1..i}, Y_{1..j}\right) = m\acute{a}x\{SCM\left(X_{1..i}, Y_{1..j-1}\right), SCM\left(X_{1..i-1}, Y_{1..j}\right)\}}$

Subsecuencia común máxima

		0	1	2	3	4	5	6	7	8
			1	0	0	1	0	1	0	1
0		0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
2	1	0	1	1	1	2	2	2	2	2
3	0	0	1	2	2	2	3	3	3	3
4	1	0	1	2	2	3	3	4	4	4
5	1	0	1	2	2	3	3	4	4	5
6	0	0	1	2	3	3	4	4	5	5
7	1	0	1	2	3	4	4	5	5	6
8	1	0	1	2	3	4	4	5	5	6
9	0	0	1	2	3	4	5	5	6	6

Subsecuencia común máxima

- Una vez que se la longitud, ¿Cómo puedo saber cuál es la subsecuencia común de dicha longitud?
  - Opción 1: a la vez que relleno la tabla con las soluciones a los problemas de dimensión menor, relleno otra tabla (o en la misma añado otro campo) que indique , para cada solución, a partir de qué subproblema se ha resuelto
    - Dependiendo del orden de las direcciones, puede cambiar la solución
  - Opción 2: a través de la tabla, bien por filas o bien por columnas

Subsecuencia común máxima

- Opción 1: rellenar otra tabla (diagonal, superior, izquierda)

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

		1	0	0	1	0	1	0	1
0		S	D	D	I	D	I	D	I
1		D	S	S	D	I	D	I	D
0		S	D	D	S	D	I	D	I
1		D	S	S	D	S	D	I	D
1		D	S	S	D	S	D	S	D
0		S	D	D	S	D	S	D	S
1		D	S	S	D	S	D	S	D
1		D	S	S	D	S	D	S	D
0		S	D	D	S	D	S	D	S

Subsecuencia común máxima

- Opción 1: rellenar otra tabla (diagonal, superior, izquierda)

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

		1	0	0	1	0	1	0	1
0		S	D	D	I	D	I	D	I
1		D	S	S	D	I	D	I	D
0		S	D	D	S	D	I	D	I
1		D	S	S	D	S	D	I	D
1		D	S	S	D	S	D	S	D
0		S	D	D	S	D	S	D	S
1		D	S	S	D	S	D	S	D
1		D	S	S	D	S	D	S	D
0		S	D	D	S	D	S	D	S

Subsecuencia común máxima

- Opción 2: a partir de la tabla
  - Por filas

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

Subsecuencia común máxima

- Opción 2: a partir de la tabla
  - Por columnas

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

Índice

- Introducción
- Problemas
  - Sucesión de Fibonacci
  - Coeficientes binomiales
  - Subsecuencia común máxima
  - Mochila 0-1
  - El problema del cambio
  - Transformación de cadenas
  - Caminos mínimos entre pares de vértices cualesquiera
  - Multiplicación de matrices

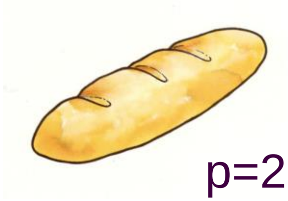
Mochila 0-1

- Dados n elementos $e_1, e_2, \ldots, e_n$ con pesos $p_1, p_2, \ldots, p_n$ y beneficios $b_1, b_2, \ldots, b_n$, y dada una mochila con capacidad de albergar hasta un máximo de peso M, queremos encontrar aquellos elementos que tenemos que introducir en la mochila de forma que la suma de los beneficios de los elementos escogidos sea máxima
- Esto es, tenemos que encontrar los valores $(x_1, x_2, \ldots, x_n)$ con $x_1, x_2, \ldots, x_n \in \{0,1\}^n$ de forma que se maximice la cantidad $\sum_{i=1}^{n} b_i x_i$, sujeta a la restricción $\sum_{i=1}^{n} p_i x_i \leq M$
- **No se pueden fraccionar los objetos**

Mochila 0-1

M=11

p=7
b=28



p=2
b=6

p=6
b=22

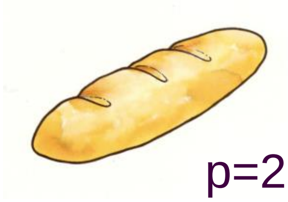
p=1
b=1

p=5
b=18

Mochila 0-1

M=11

p=7
b=28



p=2
b=6

p=6
b=22

p=1
b=1

p=5
b=18

- Introduzco los objetos:
  - Refresco + chocolate

Mochila 0-1

- ¿Qué vamos a almacenar en nuestra tabla intermedia de soluciones?
- ¿Cómo calculo la solución a mi problema basándome en soluciones a problemas parciales?
- ¿Se cumple el principio de optimalidad?

Mochila 0-1

- ¿Qué vamos a almacenar en nuestra tabla intermedia de soluciones?
- ¿Cómo calculo la solución a mi problema basándome en soluciones a problemas parciales?
- ¿Se cumple el principio de optimalidad?

- $M(i,j) =$
$$\begin{cases} 0 & si\ i = 0 \\ 0 & si\ j = 1\ y\ i < w_1 \\ b_j & si\ j = 1\ y\ i \geq w_1 \\ \max\{M(i, j-1), M\left(i - w_j, j-1\right) + b_j\} & en\ otro\ caso \end{cases}$$

Mochila 0-1

- Pesos:
  - (1, 2, 5, 6, 7)
- Beneficios:
  - (1, 6, 18, 22, 28)
- Capacidad
  - 11

	1	2	3	4	5
0	0	0	0	0	0
1	1	1	1	1	1
2	1	6	6	6	6
3	1	7	7	7	7
4	1	7	7	7	7
5	1	7	18	18	18
6	1	7	19	22	22
7	1	7	24	24	28
8	1	7	25	28	29
9	1	7	25	29	34
10	1	7	25	29	35
11	1	7	25	40	40

Mochila 0-1

- Mediante la tabla conocemos el beneficio máximo
- A partir de ella, ¿cómo obtener qué objetos hemos introducido?

	1	2	3	4	5
0	0	0	0	0	0
1	1	1	1	1	1
2	1	6	6	6	6
3	1	7	7	7	7
4	1	7	7	7	7
5	1	7	18	18	18
6	1	7	19	22	22
7	1	7	24	24	28
8	1	7	25	28	29
9	1	7	25	29	34
10	1	7	25	29	35
11	1	7	25	40	40

Mochila 0-1

- Mediante la tabla conocemos el beneficio máximo
- A partir de ella, ¿cómo obtener qué objetos hemos introducido?

	1	2	3	4	5
0	0	0	0	0	0
1	1	1	1	1	1
2	1	6	6	6	6
3	1	7	7	7	7
4	1	7	7	7	7
5	1	7	18	18	18
6	1	7	19	22	22
7	1	7	24	24	28
8	1	7	25	28	29
9	1	7	25	29	34
10	1	7	25	29	35
11	1	7	25	40	40

Índice

- Introducción
- Problemas
  - Sucesión de Fibonacci
  - Coeficientes binomiales
  - Subsecuencia común máxima
  - Mochila 0-1
  - El problema del cambio
  - Transformación de cadenas
  - Caminos mínimos entre pares de vértices cualesquiera
  - Multiplicación de matrices

El problema del cambio

- Suponiendo que el sistema monetario de un país está formado por monedas de valores $v_1, v_2, \ldots, v_n$ el problema consiste en descomponer cualquier cantidad dada M en monedas de ese país utilizando el menor número posible de monedas

- Si las monedas son de la forma $1, p, p^2, \ldots, p^n$ la solución obtenida con algoritmos voraces es óptima
- ¿Puedo obtener una solución óptima para cualquier conjunto de monedas y cualquier cantidad? (Suponemos que existe una moneda unidad) → Programación dinámica

El problema del cambio

- ¿Qué almaceno en la tabla de soluciones intermedias?
- ¿Qué ecuación de recurrencia uso utilizando la tabla?
- ¿Se cumple el principio de optimalidad?

- Ejemplo
  - Monedas: {1, 3, 7, 8}
  - Cantidad: 21

El problema del cambio

- ¿Qué almaceno en la tabla de soluciones intermedias?
- ¿Qué ecuación de recurrencia uso utilizando la tabla?
- ¿Se cumple el principio de optimalidad?

- Ejemplo
  - Monedas: {1, 3, 7, 8}
  - Cantidad: 21

- $C(i,j) = \begin{cases} 0 & si\ i = 0 \\ 1 + C(i - T[j], j) & si\ j = 1 \\ C(i, j-1) & si\ j > 1\ y\ i \leq T(j) \\ \min\{C(i, j-1), 1 + C(i - T[j], j)\} & en\ otro\ caso \end{cases}$

Índice

- Introducción
- Problemas
  - Sucesión de Fibonacci
  - Coeficientes binomiales
  - Subsecuencia común máxima
  - Mochila 0-1
  - El problema del cambio
  - Transformación de cadenas
  - Caminos mínimos entre pares de vértices cualesquiera
  - Multiplicación de matrices

Transformación de cadenas

- Sean u y v dos cadenas de caracteres. Se desea transformar u en v con el mínimo número de operaciones básicas de los siguientes tipos:
  - Eliminar un carácter
  - Añadir un carácter
  - Cambiar un carácter
- Diseñar un algoritmo que calcule el número mínimo de operaciones necesarias para transformar u en v y cuáles son esas operaciones, estudiando la complejidad en función de las longitudes de u y v.

Transformación de cadenas

- $u = abbac$
- $v = abcbc$

Transformación de cadenas

- $u = abbac$
- $v = abcbc$

- Transformación óptima – opción 1
  - Cambiar b por c
  - Cambiar a por b
- Transformación óptima – opción 2
  - Insertar c
  - Eliminar a

Transformación de cadenas

- Llamamos m a la longitud de la cadena u
- Llamamos n a la longitud de la cadena v
- $OB(m, n)$ es el número de operaciones mínimo para transformar una cadena u de longitud m en otra cadena v de longitud n

- Casos base

Transformación de cadenas

- Llamamos m a la longitud de la cadena u
- Llamamos n a la longitud de la cadena v
- $OB(m,n)$ es el número de operaciones mínimo para transformar una cadena u de longitud m en otra cadena v de longitud n

- Casos base
  - $OB(0,0) = 0$
  - $OB(m,0) = m$
  - $OB(0,n) = n$

Transformación de cadenas

- Si el último elemento de las dos cadenas es igual $\boldsymbol{u_m = v_n}$, entonces es igual al número de transformaciones con ambas cadenas menos el último elemento
  - $\boldsymbol{OB(m,n) = OB(m-1, n-1)}$
- Si el último elemento es diferente $\boldsymbol{u_m \neq u_n}$
  - Opción 1: considerar la primera cadena entera y la segunda sin el último elemento $\boldsymbol{OB(m, n-1) + 1}$
  - Opción 2: considerar la primera cadena menos el último elemento y la segunda cadena completa $\boldsymbol{OB(m-1, n) + 1}$
  - Opción 3: considerar las dos cadenas sin el último elemento $\boldsymbol{OB(m-1, n-1) + 1}$

  - Seleccionamos la opción que tenga un menor número de transformaciones

Transformación de cadenas

- $OB(m,n) = \begin{cases} n & si\ m = 0 \\ m & si\ n = 0 \\ OB(m-1,n-1) & si\ u_m = v_n \\ 1 + mín\{OB(m,n-1), OB(m-1,n), OB(m-1,n-1) & en\ otro\ caso \end{cases}$

- Complejidad: O(mn)

Transformación de cadenas

- Ejemplo
  - $u = acbbdca$
  - $v = bcabcda$

- ¿Cuál es el número mínimo de operaciones de transformación?
- ¿Cuáles son esas operaciones a partir de la tabla solución?

Índice

- Introducción
- Problemas
  - Sucesión de Fibonacci
  - Coeficientes binomiales
  - Subsecuencia común máxima
  - Mochila 0-1
  - El problema del cambio
  - Transformación de cadenas
  - Caminos mínimos entre pares de vértices cualesquiera
  - Multiplicación de matrices

Caminos mínimos

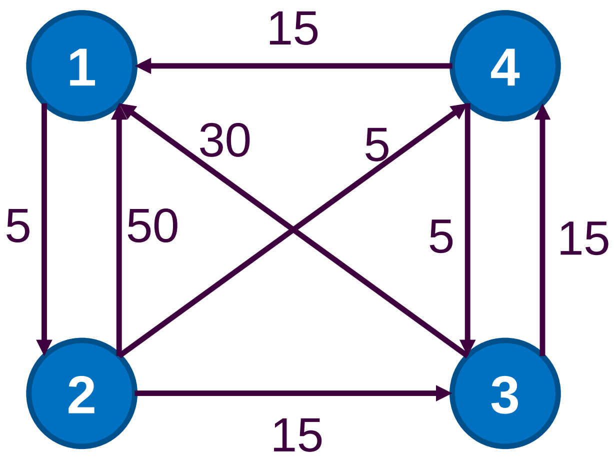
- Calcular la longitud del camino más corto entre cada par de vértices
- Dijkstra: calcular el camino más corto desde un vértice a los demás
  - Aplicar Dijkstra para cada vértice del grafo
- Otra posibilidad: algoritmo de Floyd
- Ambas soluciones tienen coste $O(n^3)$

Caminos mínimos

- Algoritmo de Floyd
  - Recorremos todos los vértices, estudiando cada uno como vértice intermedio
  - En cada caso, comparamos dos opciones:
    - El camino calculado hasta ahora
    - Ir desde el origen hasta el vértice estudiado + ir desde el vértice estudiado hasta el nodo destino
  - Principio de “optimalidad”
    - Si el camino de A a B mínimo pasa por el vértice C, entonces los caminos de A a C y de C a B también son mínimos

Caminos mínimos

- Algoritmo de Floyd

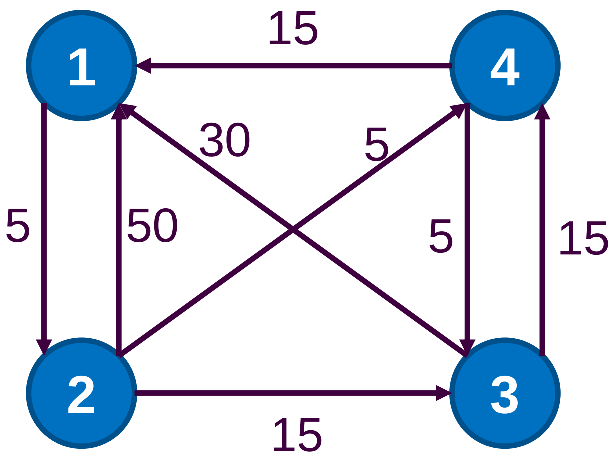


$$\begin{pmatrix} 0 & 5 & \infty & \infty \\ 50 & 0 & 15 & 5 \\ 30 & \infty & 0 & 15 \\ 15 & \infty & 5 & 0 \end{pmatrix}$$

- 1. Comenzamos con la matriz de adyacencia del grafo

Caminos mínimos

- Algoritmo de Floyd



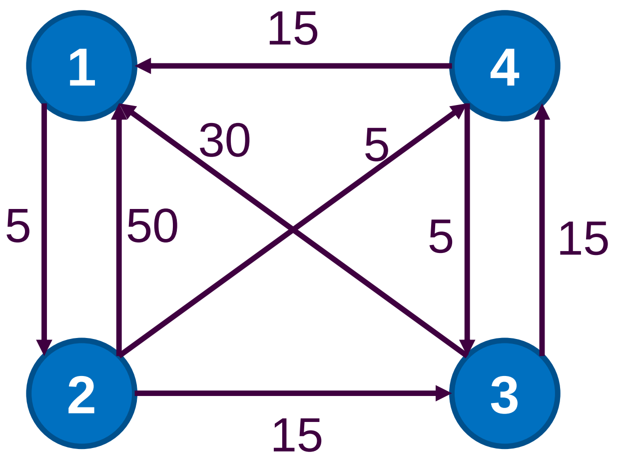
$$\begin{pmatrix} 0 & 5 & \infty & \infty \\ 50 & 0 & 15 & 5 \\ 30 & \infty & 0 & 15 \\ 15 & \infty & 5 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 5 & \infty & \infty \\ 50 & 0 & 15 & 5 \\ 30 & 35 & 0 & 15 \\ 15 & 20 & 5 & 0 \end{pmatrix}$$

- 2. Estudiamos los caminos desde origen hasta el 1 + desde el 1 a destino

Caminos mínimos

- Algoritmo de Floyd



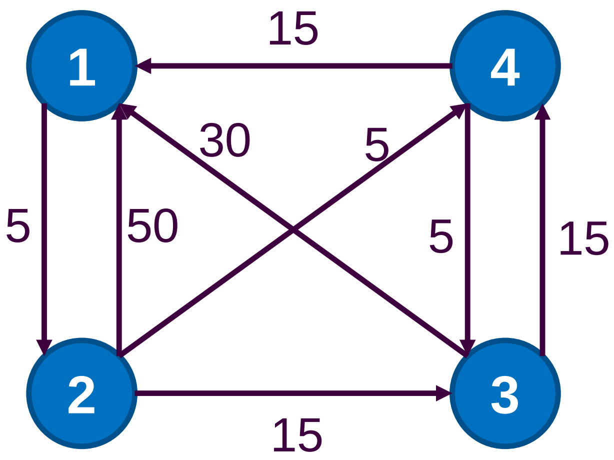
$$\begin{pmatrix} 0 & 5 & \infty & \infty \\ 50 & 0 & 15 & 5 \\ 30 & 35 & 0 & 15 \\ 15 & 20 & 5 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 5 & 20 & 10 \\ 50 & 0 & 15 & 5 \\ 30 & 35 & 0 & 15 \\ 15 & 20 & 5 & 0 \end{pmatrix}$$

- 3. Estudiamos los caminos desde origen hasta el 2 + desde el 2 a destino

Caminos mínimos

- Algoritmo de Floyd



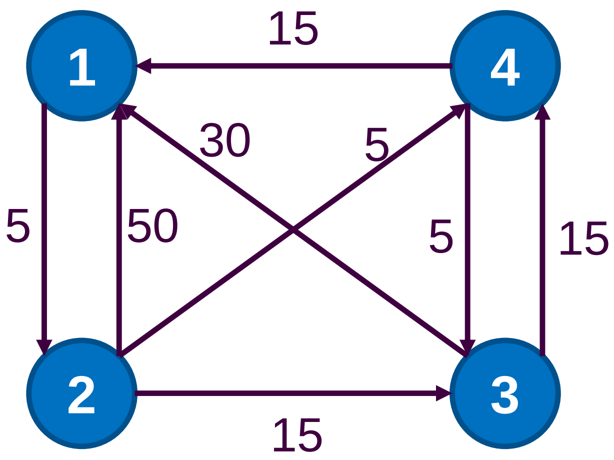
$$\begin{pmatrix} 0 & 5 & 20 & 10 \\ 50 & 0 & 15 & 5 \\ 30 & 35 & 0 & 15 \\ 15 & 20 & 5 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 5 & 20 & 10 \\ 45 & 0 & 15 & 5 \\ 30 & 35 & 0 & 15 \\ 15 & 20 & 5 & 0 \end{pmatrix}$$

- 4. Estudiamos los caminos desde origen hasta el 3 + desde el 3 a destino

Caminos mínimos

- Algoritmo de Floyd



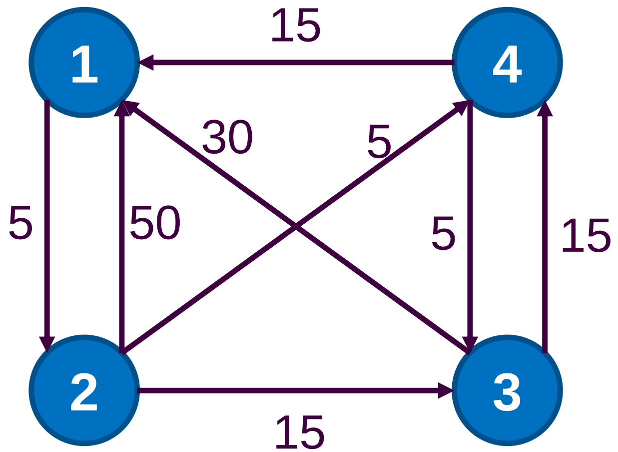
$$\begin{pmatrix} 0 & 5 & 20 & 10 \\ 50 & 0 & 15 & 5 \\ 30 & 35 & 0 & 15 \\ 15 & 20 & 5 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 5 & 15 & 10 \\ 20 & 0 & 10 & 5 \\ 30 & 35 & 0 & 15 \\ 15 & 20 & 5 & 0 \end{pmatrix}$$

- 4. Estudiamos los caminos desde origen hasta el 4 + desde el 4 a destino

Caminos mínimos

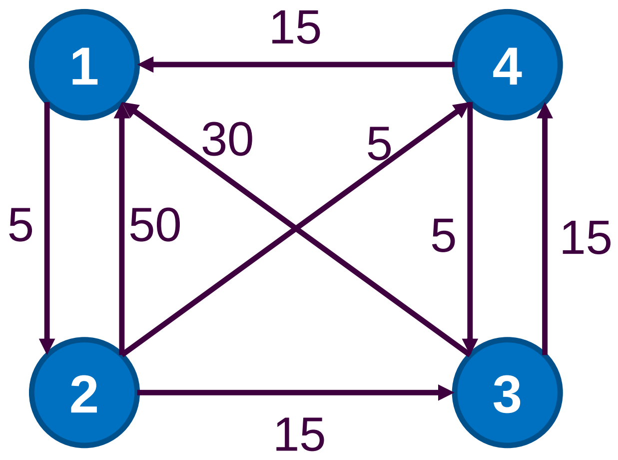
- Algoritmo de Floyd
  - ¿A partir de la tabla solución somos capaces de identificar cada uno de los caminos mínimos?¿Por qué vértices pasan?



$$\begin{pmatrix} 0 & 5 & 15 & 10 \\ 20 & 0 & 10 & 5 \\ 30 & 35 & 0 & 15 \\ 15 & 20 & 5 & 0 \end{pmatrix}$$

Caminos mínimos

- Algoritmo de Floyd
  - Necesitamos una tabla auxiliar

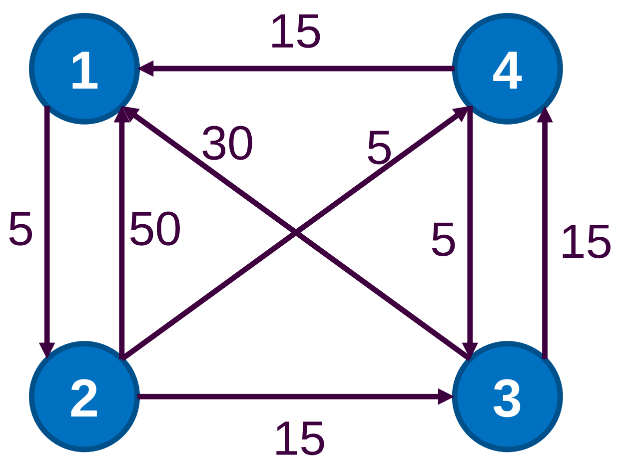


$$\begin{pmatrix} 0 & 5 & \infty & \infty \\ 50 & 0 & 15 & 5 \\ 30 & \infty & 0 & 15 \\ 15 & \infty & 5 & 0 \end{pmatrix} \Rightarrow \begin{pmatrix} 0 & 5 & \infty & \infty \\ 50 & 0 & 15 & 5 \\ 30 & 35 & 0 & 15 \\ 15 & 20 & 5 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \Rightarrow \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Caminos mínimos

- Algoritmo de Floyd
  - Necesitamos una tabla auxiliar

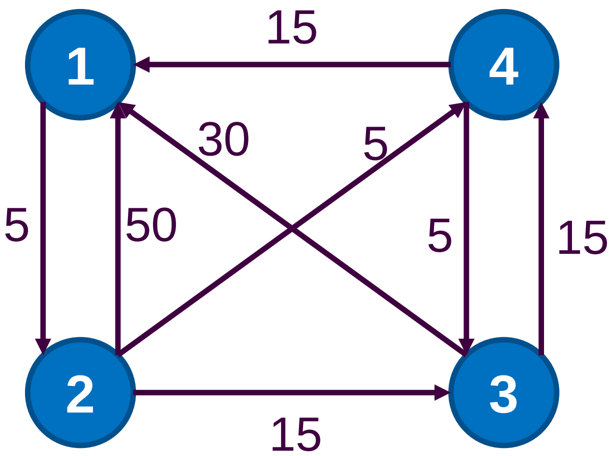


$$\begin{pmatrix} 0 & 5 & \infty & \infty \\ 50 & 0 & 15 & 5 \\ 30 & 35 & 0 & 15 \\ 15 & 20 & 5 & 0 \end{pmatrix} \Rightarrow \begin{pmatrix} 0 & 5 & 20 & 10 \\ 50 & 0 & 15 & 5 \\ 30 & 35 & 0 & 15 \\ 15 & 20 & 5 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \Rightarrow \begin{pmatrix} 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Caminos mínimos

- Algoritmo de Floyd
  - Necesitamos una tabla auxiliar

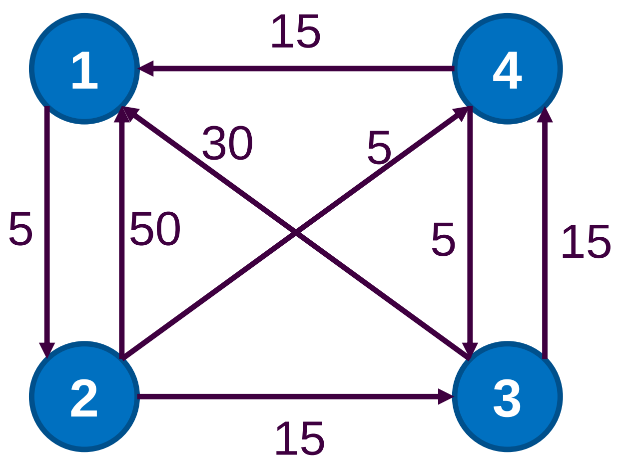


$$\begin{pmatrix} 0 & 5 & 20 & 10 \\ 50 & 0 & 15 & 5 \\ 30 & 35 & 0 & 15 \\ 15 & 20 & 5 & 0 \end{pmatrix} \Rightarrow \begin{pmatrix} 0 & 5 & 20 & 10 \\ 45 & 0 & 15 & 5 \\ 30 & 35 & 0 & 15 \\ 15 & 20 & 5 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \Rightarrow \begin{pmatrix} 0 & 0 & 2 & 2 \\ 3 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Caminos mínimos

- Algoritmo de Floyd
  - Necesitamos una tabla auxiliar

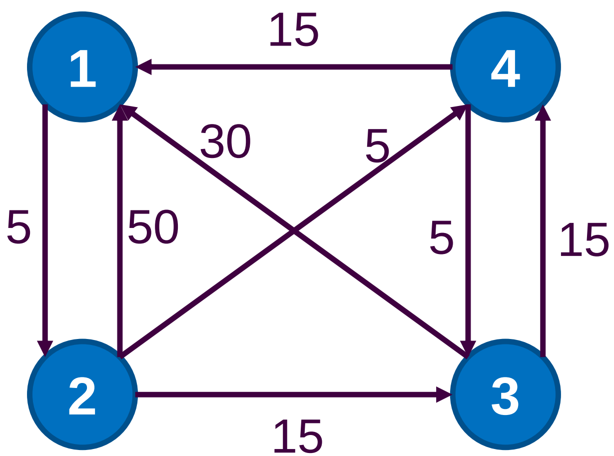


$$\begin{pmatrix} 0 & 5 & 20 & 10 \\ 50 & 0 & 15 & 5 \\ 30 & 35 & 0 & 15 \\ 15 & 20 & 5 & 0 \end{pmatrix} \Rightarrow \begin{pmatrix} 0 & 5 & 15 & 10 \\ 20 & 0 & 10 & 5 \\ 30 & 35 & 0 & 15 \\ 15 & 20 & 5 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 0 & 2 & 2 \\ 3 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \Rightarrow \begin{pmatrix} 0 & 0 & 4 & 2 \\ 4 & 0 & 4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Caminos mínimos

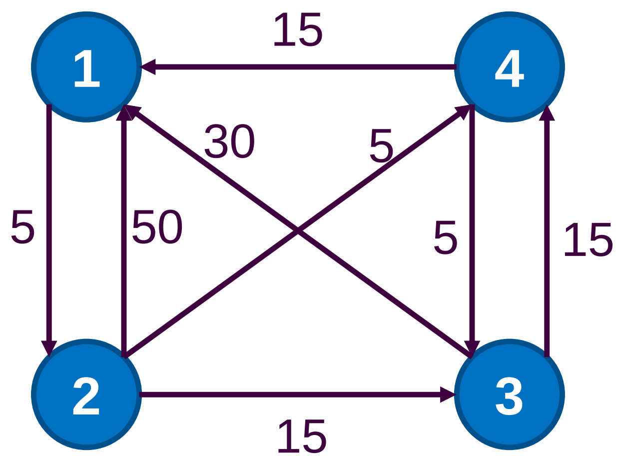
- Algoritmo de Floyd
  - ¿Por qué vértices pasa el camino del 1 al 2?
  - ¿Y del 1 al 3?



$$\begin{pmatrix} 0 & 0 & 4 & 2 \\ 4 & 0 & 4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Caminos mínimos

- Algoritmo de Floyd
  - ¿Por qué vértices pasa el camino del 1 al 2? → directo
  - ¿Y del 1 al 3? → 1 - 2 - 4 - 3



$$\begin{pmatrix} 0 & 0 & 4 & 2 \\ 4 & 0 & 4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Índice

- Introducción
- Problemas
  - Sucesión de Fibonacci
  - Coeficientes binomiales
  - Subsecuencia común máxima
  - Mochila 0-1
  - El problema del cambio
  - Transformación de cadenas
  - Caminos mínimos entre pares de vértices cualesquiera
  - Multiplicación de matrices

Multiplicación de matrices

- Queremos calcular la matriz producto M de n matrices ($M = M_1 M_2 \ldots M_n$). Por ser asociativa, existen muchas formas posibles de realizar esa operación, cada una con un coste asociado (en términos del número de multiplicaciones escalares). Si multiplico las matrices M_i de dimensiones ($a \times b$) y M_j de dimensiones ($b \times c$) se requieren abc operaciones
- Estudiar un algoritmo que obtenga siempre el menor número de multiplicaciones escalares necesarias y el orden de multiplicación de las matrices

Multiplicación de matrices

- Por ejemplo, tenemos 4 matrices, con las siguientes dimensiones: $M_1(30 \times 1), M_2(1 \times 40), M_3(40 \times$

Multiplicación de matrices

- Llamamos $M(i,j)$ al número mínimo de multiplicaciones escalares para el producto de $M_i M_{i+1} \dots M_j$
- La solución al problema es $M(1,n)$
- En el ejemplo, la última multiplicación puede ser:
  - $(M_1)(M_2 M_3 M_4)$
  - $(M_1 M_2)(M_3 M_4)$
  - $(M_1 M_2 M_3)(M_4)$
- En general, la solución viene dada por
  - $M(i,j) = (M_i M_{i+1} \dots M_k)(M_{k+1} M_{k+2} \dots M_j)$

Multiplicación de matrices

- Decimos que la matriz k tiene dimensiones $d_{k-1}\ x\ d_k$
- En general, la solución viene dada por
  - $M(i,j) = (M_i M_{i+1} \dots M_k)(M_{k+1} M_{k+2} \dots M_j)$
  - $M(i,j) = M(i,k) + M(k+1,j) + d_{i-1} d_k d_j$
- k puede tomar cualquier valor entre i y $j-1$
- Como buscamos el menor número de operaciones, la ecuación de recurrencia queda:
- $M(i,j) = \begin{cases} 0 & si\ i = j \\ mín_{i \le k < j}\{M(i,k) + M(k+1,j) + d_{i-1} d_k d_j\} & en\ otro\ caso \end{cases}$

Multiplicación de matrices

- Ejemplo
  - Calcular el número mínimo de multiplicaciones escalares para obtener $M_1 M_2 M_3 M_4$, cuyas dimensiones son:
    - $M_1(30 \times 1)$
    - $M_2(1 \times 40)$
    - $M_3(40 \times 10)$
    - $M_4(10 \times 25)$

Multiplicación de matrices

- Rellenamos los casos base
  - Cuando solo quiero multiplicar una matriz, el número de multiplicaciones escalares es 0

	$M_1\ (30x1)$	$M_2\ (1x40)$	$M_3\ (40x10)$	$M_4\ (10x25)$
M_1	0			
M_2		0		
M_3			0	
M_4				0

Multiplicación de matrices

- Rellenamos en diagonales, desde la principal hacia arriba
- $M_1 M_2$
  - $i \leq k < j \quad \rightarrow \quad k$ solo puede valer 1
  - $M(i,k) = M(1,1) = 0$
  - $M(k+1,j) = M(2,2) = 0$
  - $M(i,j) = M(1,2) = 0 + 0 + 30x1x40 = 1200$

	$M_1\ (30x1)$	$M_2\ (1x40)$	$M_3\ (40x10)$	$M_4\ (10x25)$
M_1	0			
M_2		0		
M_3			0	
M_4				0

Multiplicación de matrices

- Rellenamos en diagonales, desde la principal hacia arriba
- $M_2 M_3$
  - $i \leq k < j \quad \rightarrow \quad k$ solo puede valer 2
  - $M(i,k) = M(2,2) = 0$
  - $M(k+1,j) = M(3,3) = 0$
  - $M(i,j) = M(2,3) = 0 + 0 + 1x40x10 = 400$

	$M_1\ (30x1)$	$M_2\ (1x40)$	$M_3\ (40x10)$	$M_4\ (10x25)$
M_1	0	1200		
M_2		0		
M_3			0	
M_4				0

Multiplicación de matrices

- Rellenamos en diagonales, desde la principal hacia arriba
- $M_3 M_4$
  - $i \leq k < j \quad \rightarrow \quad k$ solo puede valer 3
  - $M(i,k) = M(3{,}3) = 0$
  - $M(k+1,j) = M(4{,}4) = 0$
  - $M(i,j) = M(3{,}4) = 0 + 0 + 40x10x25 = 10000$

	$M_1\ (30x1)$	$M_2\ (1x40)$	$M_3\ (40x10)$	$M_4\ (10x25)$
M_1	0	1200		
M_2		0	400	
M_3			0	
M_4				0

Multiplicación de matrices

- Rellenamos en diagonales, desde la principal hacia arriba
- $M_1 M_3$
  - $i \leq k < j \quad \rightarrow \quad k$ puede valer 1 ó 2

- $k = 1$
  - $M(i,k) = M(1,1) = 0$
  - $M(k+1,j) = M(2,3) = 400$
  - $M(i,j) = M(1,3) = 0 + 400 + 30x1x10 = 700$
- $k = 2$
  - $M(i,k) = M(1,2) = 1200$
  - $M(k+1,j) = M(3,3) = 0$
  - $M(i,j) = M(1,3) = 1200 + 0 + 30x40x10 = 13200$

	M_1 $(30x1)$	M_2 $(1x40)$	M_3 $(40x10)$	M_4 $(10x25)$
M_1	0	1200		
M_2		0	400	
M_3			0	10000
M_4				0

Multiplicación de matrices

- Rellenamos en diagonales, desde la principal hacia arriba
- $M_2 M_4$
  - $i \leq k < j \quad \rightarrow \quad k$ puede valer 2 ó 3

- $k = 2$
  - $M(i, k) = M(2,2) = 0$
  - $M(k + 1, j) = M(3,4) = 10000$
  - $M(i, j) = M(2,4) = 0 + 10000 + 1x40x25 = 11000$

- $k = 3$
  - $M(i, k) = M(2,3) = 400$
  - $M(k + 1, j) = M(4,4) = 0$
  - $M(i, j) = M(2,4) = 400 + 0 + 1x10x25 = 650$

	M_1 $(30x1)$	M_2 $(1x40)$	M_3 $(40x10)$	M_4 $(10x25)$
M_1	0	1200	700	
M_2		0	400	
M_3			0	10000
M_4				0

Multiplicación de matrices

- Rellenamos en diagonales, desde la principal hacia arriba
- M_1M_4
  - $i \leq k < j \quad \rightarrow \quad k$ puede valer 1, 2 ó 3

- $k = 1$
  - $M(i,k) = M(1,1) = 0$
  - $M(k+1,j) = M(2,4) = 650$
  - $M(i,j) = M(1,4) = 0 + 650 + 30x1x25 = 1400$
- $k = 2$
  - $M(i,k) = M(1,2) = 1200$
  - $M(k+1,j) = M(3,4) = 10000$
  - $M(i,j) = M(1,4) = 1200 + 10000 + 30x40x25 = 41200$

	M_1 $(30x1)$	M_2 $(1x40)$	M_3 $(40x10)$	M_4 $(10x25)$
M_1	0	1200	700	
M_2		0	400	650
M_3			0	10000
M_4				0

Multiplicación de matrices

- Rellenamos en diagonales, desde la principal hacia arriba
- $M_1 M_4$
  - $i \leq k < j \quad \rightarrow \quad k$ puede valer 1, 2 ó 3
- $k = 1$
  - 1400
- $k = 2$
  - 41200
- $k = 3$
  - $M(i,k) = M(1,3) = 700$
  - $M(k+1,j) = M(4,4) = 0$
  - $M(i,j) = M(1,4) = 700 + 0 + 30x10x25 = 8200$

	$M_1\ (30x1)$	$M_2\ (1x40)$	$M_3\ (40x10)$	$M_4\ (10x25)$
M_1	0	1200	700	
M_2		0	400	650
M_3			0	10000
M_4				0

Multiplicación de matrices

- Rellenamos en diagonales, desde la principal hacia arriba
- $M_1 M_4$
  - $i \leq k < j \quad \rightarrow \quad k$ puede valer 1, 2 ó 3
- $k = 1$
  - 1400
- $k = 2$
  - 41200
- $k = 3$
  - $M(i,k) = M(1,3) = 700$
  - $M(k+1,j) = M(4,4) = 0$
  - $M(i,j) = M(1,4) = 700 + 0 + 30x10x25 = 8200$

	$M_1\ (30x1)$	$M_2\ (1x40)$	$M_3\ (40x10)$	$M_4\ (10x25)$
M_1	0	1200	700	1400
M_2		0	400	650
M_3			0	10000
M_4				0

Multiplicación de matrices

- Si queremos saber la asociación óptima de matrices para obtener este número mínimo, necesitamos almacenar el valor de k para cada M(i,j). Podemos utilizar otra tabla del mismo tamaño.
- Complejidad $O(n^3)$