

Práctica 6.b - Ejercicios tipo recuperatorio

IMPORTANTE: TODOS los ejemplos son meramente ilustrativos, las soluciones planteadas deben ser válidas para cualquier ejemplo. Considerar en las matrices ordenadas, dicha propiedad al recorrerlas. Aplicar TODAS las buenas prácticas vistas en la cátedra.

1. Huerta Orgánica

Una huerta orgánica desea combatir las malezas con un innovador sistema robótico. Dicho sistema utiliza inteligencia artificial para detectar la especie de cada ejemplar vegetal que crece en la huerta. La huerta está organizada en N surcos. El sistema recorre cada surco y registra las especies a las que pertenece cada ejemplar vegetal que encuentra a su paso. Por cada ejemplar vegetal, registra, en una matriz V de $N \times M$, su especie, utilizando un carácter alfabético minúscula o mayúscula. Las plantas cultivadas se identifican con letras minúsculas mientras que el resto de la vegetación, tanto las plantas beneficiosas como las malezas, se identifican con letras mayúsculas. A modo de ejemplo, la siguiente matriz representa una huerta de 3 surcos de longitud 20, en cuyo primer surco se cultiva Solanum lycopersicum (tomate, 't') y Ocimum basilicum (albahaca, 'a'). Además, se pueden ver otras plantas identificadas: Calendula officinalis (calendula, 'C'), Ocimum basilicum (chamico, 'M') y Urtica dioica (ortiga, 'O'). Además, cada fila comienza y termina con uno o más caracteres 'x' para facilitar el procesamiento (no corresponde a ninguna especie).

x	t	a	C	M	O	t	a	a	t	O	C	t	t	a	O	M	C	t	x	
x	r	r	r	r	C	C	O	O	r	r	C	r	G	G	G	r	r	x	x	x
x	m	G	m	h	h	L	G	G	O	h	h	m	m	m	O	B	M	C	x	x

Para eliminar las malezas, el robot posee un láser que destruye las plantas indicadas. Para preservar los cultivos, el láser sólo puede accionarse cuando existen más de R especies vegetales no cultivadas consecutivas. No obstante, en estas situaciones, no todas las plantas no cultivadas son destruidas, sino sólo aquellas que no son beneficiosas para la huerta. Para ello, se cuenta con un arreglo B (de tamaño MAXB) que indica al sistema cuáles se consideran plantas beneficiosas.

Se pide implementar un programa en Java que indique cómo quedaría la huerta, representada por una matriz V, luego de aplicar el láser del robot, considerando un arreglo de plantas beneficiosas B y un valor R. Además, **se debe informar la cantidad total de malezas eliminadas en toda la huerta.**

Continuando con el ejemplo anterior, dado $B=\{\text{'C', 'O', 'L'}\}$ de tamaño $\text{MAXB} = 3$ y un valor $R=2$, la matriz V resultante sería:

x	t	a	C	O	t	a	a	t	O	C	t	t	a	O	C	t	x	x	x
x	r	r	r	C	C	O	O	r	r	C	r	r	r	x	x	x	x	x	x
x	m	G	m	h	h	L	O	h	h	m	m	o	c	x	x	x	x	x	x

En este ejemplo, la cantidad total de malezas eliminadas fue 9.

2. Dispositivo fitness

Un dispositivo fitness **almacena día a día**, en una matriz de NxM, los **entrenamientos de running** que realiza un atleta. Cada entrenamiento consiste en realizar **pasadas de K kilómetros** registrando cada 1 km el tiempo **en segundos que tardó**. De cada entrenamiento **se registra la actividad en secuencias separadas por cero/s**, **cada secuencia representa una pasada**. Por ejemplo, en la siguiente matriz, el sexto y séptimo día el atleta hizo 1 sola pasada, el tercer día realizó 5 pasadas (secuencias) y en el resto de los días, 4 pasadas en cada uno. También se puede ver que en la primera pasada del primer día de entrenamiento el atleta recorrió 4 km, tardando 300, 298, 298 y 297 segundos en cada kilómetro recorrido.

Por otra parte, el entrenador brinda al atleta la planificación semanal donde especifica los días (se numeran del 1 al 7) en los cuales debe hacer un entrenamiento progresivo (el arreglo es rellenado con 0 cuando no hay más días indicados). Un entrenamiento progresivo consiste en realizar pasadas en donde cada kilómetro es recorrido a mayor o igual velocidad que el anterior (en menor o igual tiempo). Por ejemplo, dado un arreglo:

1	2	5	0	0	0	0
---	---	---	---	---	---	---

En este caso el entrenamiento progresivo debe ser el primer, segundo y quinto día de la semana y se debe cumplir en todos los intervalos (secuencias) de cada día. Los días no incluidos en este arreglo son de entrenamiento libre.

Con esta información, el entrenador quiere saber si el atleta cumplió o no con el entrenamiento. **Se pide realizar un programa en JAVA que dada una matriz de NxM, un arreglo de tamaño N y un porcentaje P:**

(a) elimine del arreglo aquellos días en los cuales se cumplió con el entrenamiento establecido; (b) informe al entrenador si el atleta ha cumplido con el entrenamiento en al menos P por ciento de los días que se le asignó el entrenamiento progresivo. Para este ejemplo, el atleta cumplió con el entrenamiento progresivo 2 de los 3 días (día 2 y 5), por lo que su porcentaje de cumplimiento fue de $2/3=0.66$ o 66% de lo requerido. En este caso, para un P=0.5 (50%) el atleta cumplió con lo pedido por el entrenador.

3. Encriptación de documentos

El servicio de inteligencia secreto MD5 desea implementar un sistema de encriptación de documentos ultrasecretos. Dichos documentos son almacenados en matrices de caracteres en donde cada fila representa un renglón del mismo. Todos los renglones tienen M caracteres y comienzan y terminan con uno o más espacios.

Se desea principalmente encriptar los nombres propios (comienzan con mayúscula) que aparezcan en el documento, ya que seguramente harán referencia a información sensible como ciudades, espías propios o agentes enemigos. El mecanismo de encriptación consiste en realizar una simple inversión de los caracteres y la duplicación de las vocales minúsculas sólo de los nombres propios (secuencias) presentes en el documento. **Se pide (a) implementar el mecanismo de encriptación antes descrito sobre una matriz de NxM e (b) informar, al finalizar la ejecución del mismo, la cantidad de secuencias encriptadas.**

Por ejemplo, dada la siguiente matriz:

e	l		a	g	e	n	t	e							
J	a	m	e	s		B	o		s	e					
e	n	c	u	e	n	t	r	a		e	n				
C	o	l	o	n	i	a									

Con 3 secuencias encriptadas, la matriz encriptada resultante sería:

e	l		a	g	e	n	t	e							
s	e	e	m	a	a	J		o	o	B		s	e		
e	n	c	u	e	n	t	r	a		e	n				
a	a	i	i	n	o	o	l	o	o	C					

Puede asumir que hay espacio suficiente para incorporar los caracteres que deba.

4. Food Truck

Un food truck almacena los importes de sus ventas en un evento gastronómico en una matriz de NxM, en donde cada fila representa las ventas de un día (el evento puede durar varios días). Dentro de cada día, la información es almacenada en secuencias separadas por 0 (uno o más) donde cada secuencia representa las ventas realizadas durante una determinada hora. Por ejemplo, la siguiente tabla muestra las ventas realizadas durante el feriado largo del Día de la Bandera (4 días), en el cual durante la primera hora de trabajo del primer día del evento se realizaron las siguientes ventas: \$625, \$815 y \$900.

0	625	815	900	0	562	952	300	0	365	169	254	0	0	0	0
0	958	62	57	221	0	596	623	600	0	587	889	984	0	0	0
0	0	700	257	0	0	0	854	958	388	0	954	842	925	0	0
0	988	899	874	0	254	258	652	200	0	568	958	210	0	0	0

El dueño del food truck está analizando la incorporación de más personal, por lo tanto, desea conocer si al menos en X días consecutivos, el promedio de ventas por hora, se incrementó crecientemente (en forma ascendente).

En ejemplo anterior, con un X = 2 se deberá informar que Sí hubo al menos 2 días cuyo promedio de ventas por hora se incrementó crecientemente. Esto es así porque en el día 2 (fila 1) los promedios de cada hora fueron: \$324,5; \$606,33 y \$820 respectivamente, mientras que en el día 3 (fila 2) los promedios de cada hora fueron: \$478,5; \$733,33 y \$907.

Se pide implementar en **Java** un programa que informe lo solicitado por el dueño del food truck. **IMPORTANTE:** el ejemplo es meramente ilustrativo, la solución planteada debe ser válida para cualquier matriz y valor de X. Aplicar TODAS las buenas prácticas vistas en la materia. No usar estructuras auxiliares.

5. Compresión de imágenes

Un dispositivo que toma imágenes de un fenómeno natural está corriendo sobre una plataforma tecnológica con recursos limitados. Cada imagen (representada por una matriz de NxM) está compuesta por píxeles con valores entre 0 y 255. Se desea implementar un algoritmo para comprimir aquellas porciones de la imagen distintas del color negro (0 en la escala de valores del pixel). Se pide realizar un programa en JAVA que, dada una matriz de tamaño NxM, para cada secuencia delimitada por uno o más píxeles de color negro (valor 0) con más de X repeticiones de un valor de píxel (todos los elementos de la secuencia deben ser iguales), comprima la secuencia poniendo en la primera posición el valor negado de la cantidad de ocurrencias y a continuación el valor del pixel que se repite. Deberá informar cuánto espacio de almacenamiento se ha ahorrado con este algoritmo y cuál fila tuvo una mayor compresión. Cada fila de la matriz empieza y termina con un separador 0 (color negro). Implementar usando las buenas prácticas de programación estructurada vistas en la cátedra. Ejemplo:

Matriz que tiene una fila de la imagen:

0	67	67	67	67	67	67	67	67	0	14	0	33	33	33	33	0	5	98	0
0	23	45	45	45	45	23	0	88	88	88	88	88	0	0	0	78	78	0	0
0	0	0	45	45	45	0	45	45	45	45	45	0	45	45	0	45	0	0	0

La matriz quedaría de la siguiente forma, para un X=3:

0	-8	67	0	14	0	-4	33	0	5	98	0	0	0	0	0	0	0	0	0
0	23	45	45	45	45	23	0	-5	88	0	0	0	78	78	0	0	0	0	0
0	0	0	45	45	45	0	-5	45	0	45	45	0	45	0	0	0	0	0	0

El ahorro de almacenamiento es $6+2+3+3 = 14$ enteros (int). Si cada int ocupa 4 bytes como es en JAVA, se ahorraron en total 56 bytes. La fila más comprimida fue la fila 0.

6. Encriptación de mensajes

El servicio de inteligencia de una potencia mundial desea revitalizar su sistema de encriptación de mensajes de texto. Cada texto que debe ser encriptado es almacenado en una matriz de caracteres de NxM, donde cada fila representa un renglón del mensaje.

El texto puede contener cualquier tipo de carácter: letras, dígitos, caracteres especiales o de puntuación. El sistema de encriptación sólo debe encriptar las palabras compuestas por letras (minúsculas o mayúsculas).

El mecanismo de encriptación elegido consiste en:

- Desplazar cada vocal tantas posiciones en el abecedario como caracteres tenga la palabra.
- Duplicar cada consonante.

Por ejemplo, dada la palabra “secreto”, cuya longitud es 7, debe ser encriptada como “sslccrrltv”, dado que las consonantes se duplicaron, la vocal ‘e’ se desplazó 7 posiciones pasando a ser una ‘l’ y la vocal ‘o’ pasó a ser una ‘v’. **Se pide implementar** en Java un programa que **encripte** una matriz de caracteres de NxM e **informe** la cantidad total de palabras encriptadas.

Por ejemplo, dada la siguiente matriz con el mensaje original:

-	S	u	p	e	r		s	e	c	r	e	t	o	:	-	-	-	-	-
i	i	n	o		c	o	r	t	a	r		f	o	r					
c	o	n		r	e	t	u	r	n	!	!	!	!	!	!	!	!	!	!

la matriz con el mensaje encriptado sería:

-	S	S	z	p	p	j	r	r		s	s	l	c	c	r	r	l	t	t
---	---	---	---	---	---	---	---	---	--	---	---	---	---	---	---	---	---	---	---

i	i	n	n	q		c	c	u	r	r	t	t	g	r	r		f	f	r	r	r
c	c	r	n	n		r	r	k	t	t	{	r	r	n	n	!	!	!	!	!	

y la cantidad de palabras encriptadas sería 7.

Asumir que siempre se tiene espacio suficiente para realizar las inserciones (si se pierden otros caracteres no importa) y que cada fila de texto empieza y termina siempre con uno o más caracteres que no se corresponden con una letra. Para desplazar un carácter n posiciones, puedo sumarle a un char la cantidad de posiciones que quiero desplazarlo. char $c = 'a'$; $c = (\text{char})(c + 5)$; // c pasa a almacenar ' f '

7. Descompresión de imágenes

Un dispositivo que toma imágenes de un fenómeno natural está corriendo sobre una plataforma tecnológica con recursos limitados. Cada imagen (representada por una matriz de $N \times M$) está compuesta por píxeles con valores entre 0 y 255. Se tiene implementado un algoritmo de compresión que comprime aquellas porciones de la imagen distintas del color negro (0 en la escala de valores del pixel). Dicho algoritmo procede de la siguiente manera: por cada una de las filas de la matriz, toma cada secuencia delimitada por uno o más píxeles de color negro (valor 0) con más de X repeticiones de un valor de píxel (para ser comprimida, todos los elementos de la secuencia deben ser iguales), comprime la secuencia poniendo en la primera posición el valor negado de la cantidad de ocurrencias y a continuación el valor del pixel que se repite. Cada fila de la matriz empieza y termina con uno o más píxeles negros.

Se pide implementar el algoritmo de descompresión que restablezca la matriz original. Asumir que cada fila posee suficientes lugares como para realizar la descompresión.

Ejemplo de matriz comprimida con $X = 3$:

0	-8	67	0	14	0	-4	33	0	5	98	0	0	0	0	0	0	0	0	0	0
0	0	25	25	0	-5	3	0	25	44	44	0	-4	1	0	0	0	0	0	0	0
0	44	44	44	0	-7	15	0	-4	9	0	12	0	0	0	0	0	0	0	0	0

La matriz resultante quedaría de la siguiente forma:

0	67	67	67	67	67	67	67	67	0	14	0	33	33	33	33	0	5	98	0
0	0	25	25	0	3	3	3	3	0	25	44	44	0	1	1	1	1	0	0
0	44	44	44	0	15	15	15	15	15	15	15	0	9	9	9	9	0	12	0

Además se debe informar la cantidad total de píxeles descomprimidos y la fila en la cual se encuentra la mayor cantidad de píxeles descomprimidos (la primera, si hubiese más de una). En este ejemplo, la cantidad total de píxeles descomprimidos fue 32 y la fila con más píxeles descomprimidos fue la fila 0.

8. Góndolas de supermercado

Un supermercado organiza sus góndolas utilizando una matriz de $N \times M$. La matriz representa una góndola vista de frente en donde cada fila son sus estantes y las columnas la posición donde se ubica un determinado producto. El valor almacenado en cada elemento de la matriz es el precio de cada producto. Cada estante posee separadores que son representados en la matriz con uno o más ceros.

Por ejemplo en la siguiente matriz podemos ver una góndola de 3 estantes (filas). En la primera fila, se encuentran los productos que cuestan \$120, \$250 y \$80, luego dos separadores y otros productos.

0	120	250	80	0	0	620	410	645	0	0	240	960	0	0					
0	250	155	90	85	0	150	625	0	0	900	750	225	0	0					
0	580	550	850	0	0	220	110	0	0	150	480	690	0	0					

El gerente desea ordenar el diseño de las góndolas y para ello solicitó un programa que indique cómo quedaría la góndola si los grupos (secuencias) de productos que se encuentran ordenados descendente, se invirtieran en orden ascendente. Además, desea conocer en cuáles estantes al menos se reordenaron X grupos de productos.

Siguiendo el ejemplo, la góndola debería quedar de la siguiente forma:

0	120	250	80	0	0	620	410	645	0	0	240	960	0	0
0	85	90	155	250	0	150	625	0	0	225	750	900	0	0
0	580	550	850	0	0	110	220	0	0	150	480	690	0	0

Además, debe informar, para un X = 2, que en el estante 2 (fila 1) hubo al menos 2 reordenamientos.

9. Ventas supermercado

Un supermercado almacena los importes de sus ventas anuales en una matriz de NxM, en donde cada fila representa las ventas de cada mes. Dentro de un mes, la información es almacenada en secuencias separadas por 0 donde cada secuencia representa las ventas realizadas dentro de un día. Por ejemplo, en la siguiente tabla, en el mes 1 se realizaron 3 ventas el primer día por \$150, \$200 y \$165.

0	0	150	200	165	0	154	352	240	256	0	900	750	0	0
0	940	105	265	845	215	0	245	765	348	0	741	125	541	0
0	851	543	625	845	914	0	754	184	452	637	917	0	0	0

El gerente desea solicitar distintas estadísticas según el mes que se quiere analizar. En algunos casos le interesa conocer el **promedio mensual de las ventas de mayor importe registradas en cada día** y en otros casos, necesita saber si el **promedio diario de ventas fue todos los días superior a X durante el mes**. Para ello informa en un arreglo A1 los meses que desea la primera estadística y en un arreglo A2 los de la segunda. A1 y A2 son de tamaño N, llenados con 0.

Por ejemplo, si A1 = {1,2,0} y A2 = {3,0,0} para los meses 1 y 2 se informará que el promedio es \$484 (200 + 352 + 900 / 3) y 815.33 (940 + 765 + 741 / 3), respectivamente. Para el mes 3, dado un X = 600, informará que no se cumplió.