

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA
ALGORITMIA

2da. práctica (tipo B)
(Primer Semestre 2023)

Duración: 2h 50 min.

- En cada función el alumno deberá incluir, a modo de comentario, la estrategia o forma de solución que utiliza para resolver el problema. De no incluirse dicho comentario, el alumno perderá el derecho a reclamo en esa pregunta.
- No puede emplear STL, Plantillas o funciones no vistas en los cursos de la especialidad.
- Los programas deben ser desarrollados en el lenguaje C++. Si la implementación es diferente a la estrategia indicada o no la incluye, la pregunta no será corregida.
- Un programa que no muestre resultados coherentes y/o útiles será corregido sobre el 50% del puntaje asignado a dicha pregunta.
- Debe utilizar comentarios para explicar la lógica seguida en el programa elaborado. El orden será parte de la evaluación.
- Se utilizarán herramientas para la detección de plagios, por tal motivo si se encuentran soluciones similares, se anulará la evaluación a todos los implicados y se procederá con las medidas disciplinarias dispuestas por la FCI.
- Para esta evaluación solo se permite el uso de las librerías **iostream**, **iomanip**, **cmath**, **fstream** y **cstring**
- Su trabajo deberá ser subido a PAIDEIA.
- Los archivos deben llevar como nombre su código de la siguiente forma **codigo_LabX_PY**

Pregunta 1 (10 puntos)

Los antivirus ayudan a prevenir la infección por malware en los computadores, así como también realizan la detección y eliminación de amenazas de seguridad evitando así posibles impactos negativos a las empresas por incidentes de seguridad. Una de las funciones por defecto con las que cuentan los antivirus es el análisis en tiempo real, es decir que compara, con una base de datos de firmas de virus conocidos y patrones de comportamiento malicioso, los archivos que se están leyendo o ejecutando en el computador. Cuando son virus conocidos y encuentra la coincidencia en la ejecución del archivo, el antivirus detiene la ejecución, advierte al usuario y le solicita qué acción realizar: eliminar o poner en cuarentena el archivo. Cuando son patrones de comportamiento malicioso, analiza el archivo para detectar patrones sospechosos de comportamiento. Es así que, al realizar el análisis instrucción por instrucción, registra el número de registro del posible código malicioso y la cantidad de líneas de instrucción de posible código malicioso. Este análisis genera 2 matrices $N \times M$, una de ellas tiene el número de registro (ID) de los posibles códigos maliciosos y la segunda matriz la cantidad de líneas de instrucciones que coinciden con los códigos maliciosos. Un ejemplo de este análisis es el siguiente: Matrices $N=5$, $M=6$

<i>matriz_registro</i>						<i>matriz_lineas_instruccion</i>					
-1	-2	-3	-4	-5	-6	10	5	8	9	8	1
-7	-8	1	-9	2	-10	10	5	2	8	2	3
-11	1	2	-12	-13	-14	4	2	1	4	8	1
-15	-16	-17	1	-18	-19	1	2	4	3	5	3
-20	1	2	-21	-22	-23	8	3	3	8	5	3

Un extracto de estas matrices se interpreta de la siguiente manera:

matriz_registro	...	matriz_lineas_instruccion	...	
	-8		5	5 instrucciones que no refieren amenaza de seguridad
	1		2	2 instrucciones que son posible código malicioso. Coinciden con el virus
	-9		8	8 instrucciones que no refieren amenaza de seguridad
	2		2	2 instrucciones que son posible código malicioso. Coinciden con el virus
	-10		3	3 instrucciones que no refieren amenaza de seguridad
	-11		4	4 instrucciones que no refieren amenaza de seguridad
	1		2	2 instrucciones que son posible código malicioso. Coinciden con el virus
	2		1	1 instrucciones que son posible código malicioso. Coinciden con el virus
	-12		4	4 instrucciones que no refieren amenaza de seguridad
	-13		8	8 instrucciones que no refieren amenaza de seguridad
	

Es así como, por cada análisis que realiza el antivirus, si existen en total 10 o más líneas de instrucciones de un mismo virus, se envía una alerta al usuario. Siguiendo con el ejemplo dado:

- Para el virus de registro 1, se obtienen 10 coincidencias en total.
 - Para el virus de registro 2, se obtienen 6 coincidencias en total.
 - Por lo que se considera que el archivo cuenta con código malicioso, específicamente del virus de registro 1.
- a) **(1 punto)** Desarrolle las sentencias y estructuras necesarias para el ingreso de datos.
- b) **(9 puntos)** Implemente un algoritmo utilizando la estrategia divide y vencerás que, ingresando el registro de un determinado virus encuentre **desde las 2 matrices**, la cantidad de líneas de instrucciones identificadas por el antivirus en el análisis en tiempo real y determine si el archivo tiene o no posible código malicioso del virus con el registro ingresado.

*Si en caso se requiere un proceso de ordenamiento o búsqueda, **debe usar la estrategia de divide y vencerás. Sólo puede usar arreglos unidimensionales para el ordenamiento de los datos de las matrices.***

Pregunta 2 (10 puntos)

Una empresa vitivinícola debe mover las barricas de vino de una bodega a otra para mejorar el proceso de maduración, usualmente la bodega destino es del mismo tamaño o de menor capacidad que la bodega origen. Para realizar esta tarea adquiere un robot Wall-Ye Win, esta unidad solo maneja algoritmos recursivos, no puede trabajar con sentencias iterativas y su memoria interna solo opera con 2 matrices (arreglos bidimensionales) y variables enteras. Para un mejor trabajo una matriz puede ser empleada para representar la bodega origen y la otra la bodega destino.

Se sabe que dentro de los almacenes las barricas están apiladas por separado de acuerdo con el año de la cosecha, además al apilarlas las barricas de mayor peso van en la parte inferior y la de menos peso en la parte superior para evitar romper los depósitos, los pesos son de 100 kg, 50 kg y 25 kg. Se sabe que en el almacén de origen las pilas están en desorden en cuanto al año de producción, pero en el almacén de destino deben colocarse las barricas más antiguas al fondo. A continuación, un ejemplo de datos de ingreso:

Almacén origen:

Pila 1	2010-100	2010-50	2010-50		
Pila 2	2021-100	2021-100	2021-50	2021-50	
Pila 3	2018-50	2018-50	2018-25	2018-25	2018-25
Pila 4	2015-100	2015-50	2015-25		
Pila 5	2020-100	2020-100			
Pila 6	2019-100	2019-050	2019-050	2019-25	
Pila 7	2022-100	2022-100			

(las pilas se representan como filas, las más pesadas al lado izquierdo)

Máximo de barricas que pueden apilarse $c = 5$

Número de pilas dentro del almacén $p = 7$

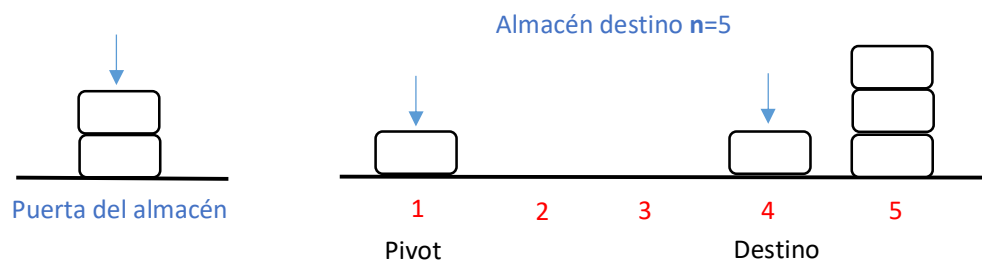
Almacén destino:

Máximo de barricas que pueden apilarse $m = 4$

Número de pilas dentro del almacén $n = 5$

El proceso de mover las pilas de un almacén a otro es el siguiente:

- El robot selecciona la pila más antigua del almacén de origen que pueda entrar en el almacén destino. El número de barricas debe ser menor o igual a m (máximo de barricas que pueden apilarse en el almacén de destino).
- Colocar la pila completa seleccionada sin desordenarla en la puerta del almacén de destino, esta operación es posible ya que el robot tiene la fuerza para levantar la pila completa y el almacén de origen brinda el espacio para esta operación.
- Ahora el robot debe mover la pila que está en la puerta y colocarla lo más al fondo posible dentro del almacén de destino, pero debido a su tamaño y estructura, el robot no puede levantar la pila completa, sino que debe mover cada barrica a la posición final dentro del almacén, pero sin aplastar las que tienen menor peso. Para esta tarea pueden usar el primer espacio de apilamiento en el almacén de destino como posición pivot, lo que origina que este espacio al final del proceso siempre quede libre y la capacidad del almacén de destino sea realmente $n-1$ pilas.



De acuerdo con el ejemplo inicial, el almacén de destino debe quedar de la siguiente forma:

Pila 1				
Pila 2	2020-100	2020-100		
Pila 3	2019-100	2019-50	2019-50	2019-25
Pila 4	2015-100	2015-50	2015-25	
Pila 5	2010-100	2010-50	2010-50	

Como observa la ubicación para la pila 1 esta siempre vacía, aunque quedan pilas por mover.

(10 puntos) Implemente un programa en C++ que utilice una función recursiva que se encargue de trasladar las barricas de un almacén a otro. No puede usar iteraciones, ya que anularía su pregunta, excepto para imprimir el contenido del almacén destino como respuesta final. Recuerde que debe mover las barricas de un almacén a otro no copiarlo. Las pilas que no se mueven deben quedar en el almacén de origen. Solo debe usar variables enteras y dos matrices. **No puede usar arreglos extras o estructuras.**

Sugerencia: Para manejar la pila que se coloca en la puerta del almacén de destino puede agregar una fila más en la matriz que la representa, pero solo temporalmente.

Profesores del curso:

David Allasi
Rony Cueva
Fernando Huamán

San Miguel, 06 de mayo del 2023