PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA

ALGORITMIA

Primer Examen

(Segundo Semestre 2022)

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.
- 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. Tampoco está permitido usar código obtenido en internet.
- Para este examen solo se permite el uso de las librerías **stdio.h, stdlib.h** y **math.h**
- **No están permitidas las funciones para obtener el tamaño de la pila o cola que recorran estas estructuras.**
- Su trabajo deberá ser subido a PAIDEIA.
- Los archivos deben llevar como nombre su código de la siguiente forma codigo_EX1_P# (donde # representa el número de la pregunta a resolver)

Resuelva solo 2 de las siguientes preguntas:

Pregunta 1 (10 puntos)

La empresa de pollos a la brasa Tiko's ha tenido un éxito inesperado en sus ventas, por tal motivo ha decidido ampliar su cantidad de sucursales. Debido a este incremento la cantidad de pedidos de pollos crudos como insumo principal ha aumentado, por tal motivo el área de compras ha ampliado los días que recibe pedidos de cada sucursal, ahora es de lunes a viernes, de tal forma que pueda programar todas las compras los sábados y enviarlos a avícola "San Bernardo". Se sabe que cada día el área de compras recibe un conjunto de pedidos de las diferentes sucursales, en donde se registran la hora en la que quieren recibir sus pedidos el domingo y el número de la sucursal solicitante. Al término de cada día, el área de compras ordena ascendentemente los pedidos realizados en ese día con respecto a la hora indicada. De esa forma, el área de compras cuenta con una lista de pedidos ordenados por hora: una lista recibida cada día.

A continuación, se muestra un ejemplo cómo estarían ordenados los pedidos al llegar al sábado:

- Lunes: 8(6)->10(14)->12(1)
- Martes: 9(3)->11(8)
- Miércoles: 8(2)->9(5)->10(10)
- Jueves: 14(13)->15(9)->16(11)
- Viernes: 17(4)->18(12)->19(7)

Cada nodo se representa de la siguiente forma: hora(sucursal)

El sábado a primera hora, el área de compras organiza las entregas que realizará a "San Bernardo" durante el domingo, por lo que requiere obtener una sola lista ordenada de todos los pedidos realizados en la semana. Para construir esta lista ordenada debe tener en cuenta lo siguiente:

- Almacenar la información de los pedidos de cada día en una lista simplemente enlazada.
- Si dos pedidos tienen la misma hora de entrega, se debe colocar el pedido que fue hecho primero en la lista ordenada. Por lo tanto, un pedido que se hizo el lunes para ser entregado a las 8 debe ir antes que un pedido realizado el miércoles para ser entregado también a las 8.
- Cuando se tienen que fusionar dos listas con horarios disjuntos (el caso de jueves y viernes, en el ejemplo anterior), la fusión debería realizarse en tiempo constante, complejidad *O*(1).
- En el peor de los casos, la fusión de dos listas debe tener complejidad *O*(n).
- El algoritmo de fusión no puede emplear **memoria extra** (es decir, el **algoritmo de fusión no puede usar malloc en ninguna parte**), **tampoco estructuras adicionales, TAD's, arreglos o matrices**.

Se solicita lo siguiente:

a) Defina las estructuras de datos que soporten la aplicación e implemente el ingreso de los datos en las 5 listas que corresponden a los días de lunes a viernes y finalmente ordene estas listas de acuerde al caso (2.0 puntos)
b) Implemente el algoritmo de fusión de listas (7.0 puntos)
c) Muestre la lista final donde para cada elemento se muestre la hora de entrega, sucursal y el día en el que el pedido fue realizado (1 punto)

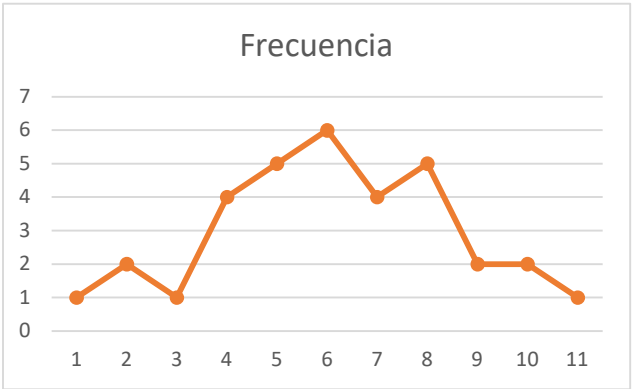
La salida para el ejemplo dado sería así:
8(6) (Lunes) -> 8(2)(Miercoles) -> 9(3)(Martes) -> 9(5)(Miercoles) -> 10(14)(Lunes) -> 10(10) (Miercoles) -> 11(8)(Martes) -> 12(1)(Lunes) -> 14(13)(Jueves) -> 15(9)(Jueves) -> 16(11)(Jueves) -> 17(4)(Viernes) -> 18(12)(Viernes) -> 19(7)(Viernes)

Pregunta 2 (10 puntos)

Un famoso cantante de música urbana llamado "Bad Sunny" está de gira por nuestro país, pero por error olvido su sistema auto tune que le "ayuda" a no desafinar al cantar. Debido a este problema el cantante ha decidido cancelar su concierto, ya que sin el sistema no puede dar su espectáculo. Por tal motivo la empresa que lo ha contratado para la gira ha decidido buscar una alternativa para ayudar al reguetonero a no desafinar. Luego de una intensa búsqueda encuentran un robot musical llamado DJ Robot, esta unidad es muy avanzada y puede manejar pilas y colas que le permiten realizar diferentes operaciones, con el fin de realizar correcciones musicales en tiempo real específicamente de las frecuencias altas de los cantantes poco entonados.

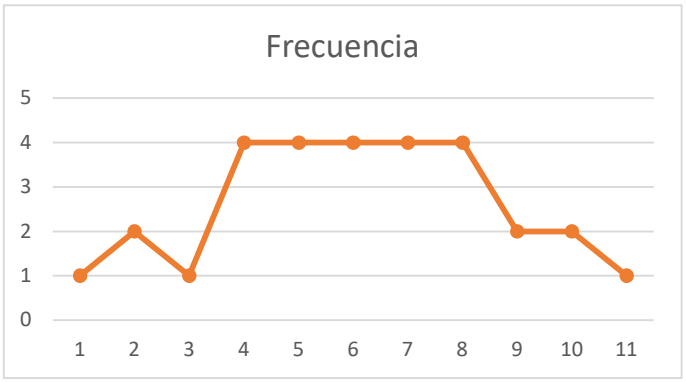
A continuación, se muestra un ejemplo de una corrección musical que debe realizar el robot, a partir de una secuencia de frecuencias musicales, similar a que se muestra a continuación:

Tiempo	Frecuencia
1	1
2	2
3	1
4	4
5	5
6	6
7	4
8	5
9	2
10	2
11	1



El algoritmo que usa el robot debe detectar la frecuencia mas alta que dure mayor cantidad de tiempo, ya que durante ese tiempo es donde el reguetonero desafina, y se deben nivelar las frecuencias, para el ejemplo anterior la frecuencia mas alta que dura mas tiempo es de 4, y ocurre a partir del tiempo 4 hasta el tiempo 8, en otras palabras, dura 5 tiempos.

Tiempo	Frecuencia
1	1
2	2
3	1
4	4
5	4
6	4
7	4
8	4
9	2
10	2
11	1



Desarrolle un algoritmo que ayude al robot a eliminar las frecuencias en las que desafina el "cantante" urbano, calculando la **frecuencia mas alta que dure mas tiempo**, así como el **tiempo que dura**. Para esta tarea solo puede usar una pila o una cola. Los datos de entrada ingresan al robot mediante un arreglo (frecuencias). Debido a que el proceso debe ser en tiempo real, la complejidad del algoritmo debe ser ***O***(n). El uso de otras estructuras no indicadas en la pregunta invalida la respuesta.
Para el ejemplo anterior la salida será: **Frecuencia máxima=4** **Duración=5**

PREGUNTA 3 (10 puntos)

La criba de Eratóstenes es un algoritmo que permite hallar todos los números primos menores que un número natural "n". Se forma una tabla con todos los números naturales comprendidos entre 2 y n, y se van **tachando** los números que no son primos de la siguiente manera (algoritmo):

- Comenzando por el número 2 (número procesado) se inicia el proceso y se tachan (eliminan) todos sus múltiplos.
- Comenzando de nuevo, después del último número procesado, se busca un número entero que no ha sido tachado y si ese número es primo, se procede a tachar (eliminar) todos sus múltiplos, y así sucesivamente se repite.
- El proceso termina cuando el cuadrado del mayor número confirmado como primo (número a procesar) es mayor que n. En ese momento imprime todos los números que quedan (no han sido tachados).

Los números del 2 al "n" se guardarán en una **lista** como la que se muestra en la figura 1, en la cual se toma como ejemplo un n = 15.

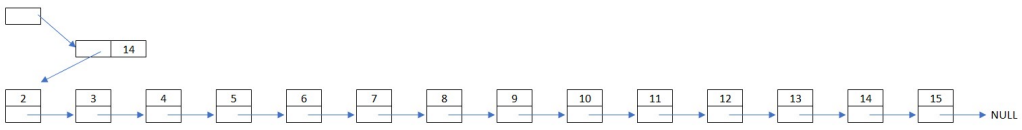


Figura 1

Conforme va ejecutando el algoritmo descrito, por cada número que va procesando debe ir

formando una cola donde vaya colocando los números que va tachando (eliminando) de la lista para dicho número. Por ejemplo, para el número 2 irá formando la siguiente **cola** (ver figura 2).

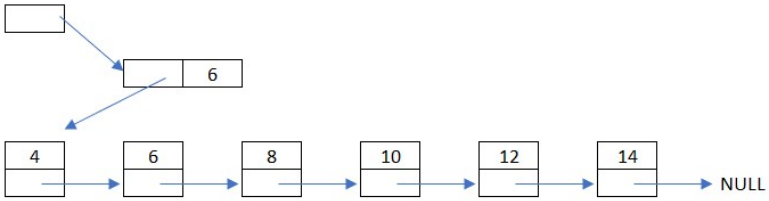


Figura 2

A la vez que se va generando la cola de la figura 2, también se debe ir actualizando la **lista** de la figura 1, eliminando los nodos de los números que se van tachando. Al realizar esta operación, la lista quedaría al final de la siguiente manera. (ver figura 3)

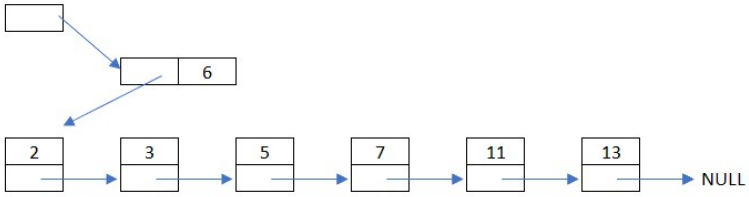


Figura 3

Conforme va desarrollando el algoritmo, debe imprimir un reporte, similar al que se muestra, con la información del proceso ejecutado por la criba. Este reporte se debe obtener recorriendo la lista de la figura 1 y la cola de la figura 2.

Proceso de Criba de Eratóstenes:

Número Procesado: 2
Números Tachados para el 2: 4 -> 6 -> 8 -> 10 -> 12 -> 14.

Número Procesado: 3
Números Tachados para el 3: 9 -> 15.

Número Procesado: 5, no tiene números tachados. Número Procesado: 7, no tiene números tachados. Número Procesado: 11, no tiene números tachados. Número Procesado: 13, no tiene números tachados.

Se le pide elaborar un programa en C que permita leer un número n, ejecute el algoritmo presentado para la criba de Eratóstenes, siguiendo estrictamente cada uno de los pasos utilizando las estructuras indicadas en la figura 1 y figura 2 y por último muestre el reporte indicado. Para esta tarea realice:

a) Defina las estructuras necesarias que soporte lo solicitado en el algoritmo descrito (1 punto).
b) Implemente una función que permita generar la lista inicial de la figura 1 en base a un número n (2 puntos).
c) Implementar una función recursiva que permita desarrollar el algoritmo descrito, imprimiendo el reporte solicitado al final (7 puntos).

Para el desarrollo de esta pregunta no puede usar arreglos o alguna TAD extra que no se encuentra indicado en el enunciado. Sino emplea la estrategia indicada invalida su respuesta.

Profesores del curso:

David Allasi
Rony Cueva

San Miguel, 15 de octubre del 2022