



2024-I

UNAM

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN

MATERIA

ESTRUCTURA DE DATOS

TEMA

Implementar una Cola

PROFESOR

Mtro. ISC. MIGUEL ANGEL SÁNCHEZ HERNÁNDEZ

Introducción

La Cola es otro tipo especial de lista, en el cual los elementos se insertan en un extremo (cabeza) y se suprimen en el otro (cola). Estas también las conocemos como FIFO (first-in, first-out), primero en entrar, primero en salir.

Las operaciones que podemos realizar con la estructura de la Cola, son las siguientes:

- `borrar()`: Borra la Cola.
- `esVacia()`: Regresa un booleano, verdadero si esta vacía.
- `insertar()`: Ingresa un elemento a la Cola (por la cabeza).
- `extraer()`: Regresa un elemento de la Cola (por la cola).
- `primerElemento()`: Regresa el primer elemento de la Cola sin eliminarlo.

Ejercicio: Ocupando la clase de Lista, implementa la clase de Cola.

Ejercicio 1: Cuando se vende una acción de una empresa, la ganancia de capital (o a veces pérdida) es la diferencia entre su precio de venta y el precio pagado originalmente para comprarla. Esta regla se puede comprender con facilidad para el caso de una sola acción, pero si se venden varias acciones compradas durante largo tiempo, se deben identificar las que realmente se venden. Un principio normal en contabilidad, para identificar cuáles acciones se vendieron en este caso, es usar un protocolo FIFO: las acciones que se venden son las que se han conservado el mayor tiempo. Por ejemplo, supóngase que se compran 100 acciones a \$20.00 en el día 1, 20 a \$24.00 en el día 2, 200 a \$36.00 en el día 3, y que se venden 150 acciones a \$30.00 en el día 4. Si aplicamos el protocolo FIFO nos queda de la siguiente manera: $(100 \cdot 10) + (20 \cdot 6) + (30 \cdot -6) = \940.00 . Escribe un programa donde se pueda leer un archivo con el siguiente formato:

`C;200;30;1`

`C;300;14;2`

`V:180;45;3`

Etc.

En donde "`C;200;30;1`" indica que se compraron 200 acciones a \$30.00 el primer día, tienes que meter estos datos en una cola, para después calcular la ganancia de capital.

Ejercicio 2: Un verdugo es mandado a exterminar a n prisioneros de guerra. El exterminio lo ejecuta de la siguiente manera: los prisioneros forman un círculo alrededor del verdugo, el verdugo elige a quien fusilar primero, una vez muerto el primero, el verdugo cuenta, a partir del lugar donde estaba su última víctima, k prisioneros en orden de las manecillas del reloj, y luego fusila al k -ésimo prisionero después de su última víctima (a los muertos no los cuenta), y repite este proceso hasta que solo quede un prisionero. El último prisionero podrá ser liberado. El verdugo tiene un amigo entre los n prisioneros, escribe un programa que dado, n , k y la ubicación de su amigo, le diga a quien fusilar primero, para asegurar que su amigo sea el que quede libre.

Ejercicio 3: El estacionamiento del centro tiene un solo carril que aloja hasta diez carros. Los automóviles llegan por el extremo sur del estacionamiento y salen por el extremo norte. Si llega un cliente para recoger un carro que no está en el extremo norte, se sacan todos los automóviles de ese lado, se retira el auto y los otros coches se establecen en el mismo orden que estaban. Cada vez que sale un auto, todos los autos del lado sur se mueven hacia adelante para que en todas las ocasiones todos los espacios vacíos estén en la parte sur del estacionamiento. Escriba un programa que emule el funcionamiento del estacionamiento. Se recomienda que genere números al azar para simular las llegadas y salidas de los carros, por ejemplo si queremos indicar que llego un auto tomamos un intervalo $[0,60]$, y salida $[61,90]$, analiza que datos tienes que almacenar para saber que auto tiene que salir.

Ejercicio 4: (Simulación de supermercado) Escriba un programa que simule una línea para pagar en un supermercado. La línea es un objeto cola. Los clientes (es decir, los objetos cliente) llegan en intervalos enteros aleatorios de 1 a 4 minutos. Además, a cada cliente se le atiende en intervalos enteros aleatorios de 1 a 4 minutos. Obviamente, los ritmos necesitan balancearse. Si el ritmo promedio de llegadas es mayor que el ritmo promedio de atención, la cola crecerá infinitamente. Incluso con ritmos “balanceados”, el factor aleatorio puede aún provocar largas líneas. Ejecute la simulación del supermercado durante un día de 12 horas (720 minutos), utilizando el siguiente algoritmo:

a) Elegir un entero aleatorio entre 1 y 4 para determinar el minuto en el que debe llegar el primer cliente.

b) Al momento en que llegue el cliente:

Determinar el tiempo de atención del cliente (entero aleatorio de 1 a 4). Empezar a atender al cliente.

Programar la hora de llegada del siguiente cliente (se suma un entero aleatorio de 1 a 4 al tiempo actual).

c) Para cada minuto del día: Si llega el siguiente cliente, hay que proceder de la siguiente manera:

Decirlo así.

Poner al cliente en la cola.

Programar la hora de llegada del siguiente cliente.

Si se terminó de atender al último cliente:

Decirlo así.

Sacar de la cola al siguiente cliente al que se va a atender.

Determinar el tiempo requerido para dar servicio al cliente (se suma un entero aleatorio del 1 al 4 al tiempo actual).

Ahora ejecute su simulación durante 720 minutos y responda a cada una de las siguientes preguntas:

a) ¿Cuál es el máximo número de clientes en la cola, en cualquier momento dado?

b) ¿Cuál es el tiempo de espera más largo que experimenta un cliente?

c) ¿Qué ocurre si el intervalo de llegada se cambia de 1 a 4 minutos por un intervalo de 1 a 3 minutos?