

Estructura de datos: Cola circular y cola con prioridad.

Cola circular

Para entender un poco más las colas circulares, veremos el siguiente video:

<https://www.youtube.com/watch?v=htcklENh-4k>

Ahora bien, las colas circulares, como podemos notar, se basan en las colas estáticas, ya que este tiene un límite de datos a entrar. Como se menciona en el video, la gran diferencia y ayuda de este tipo de colas es que no hay un gasto computacional extra para volver a recorrer los datos, es decir, poner el dato que entro después al principio, y así recorrer todos los elementos. Esto también se podría ver como un ahorro de memoria o de almacenamiento en la cola, ya que se vuelve a reciclar las posiciones para no estar recorriendo todos los datos una posición antes, después de que estén todos los datos. Veámoslo un poco gráfico de nuevo.

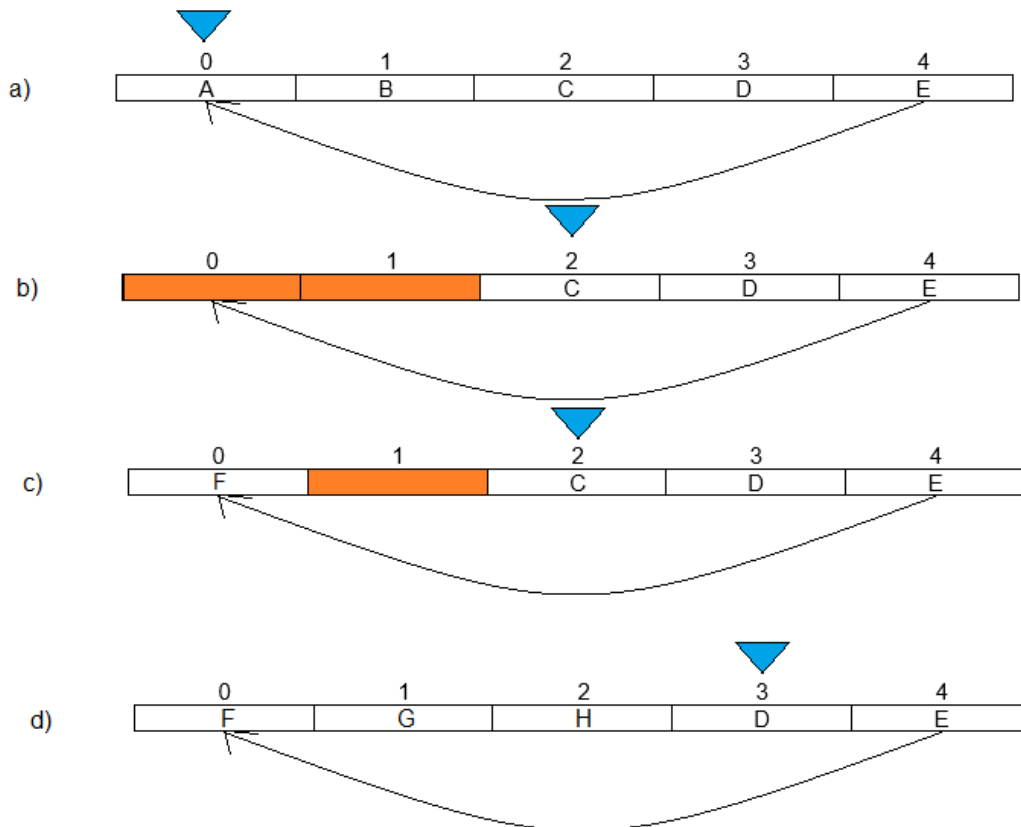


Figura 1. a) Cola llena. b) Sacar dos elementos. c) Ingresar nuevo elemento.

d) Sacar un elemento e ingresar dos

Tenemos la figura 1, donde podemos observar 4 incisos, vayamos uno por uno, En el inciso a) vemos una cola llena, donde tenemos como primer elemento a A, en la posición cero y como último elemento la posición 4 con el dato E, ahora, ¿por qué no eliminamos dos

elementos? Veamos, cuando nosotros sacamos un elemento, el índice se tiene que mover al dato que le sigue, que sería el dato B, al sacar otro dato (B), el índice pasará al dato C, posición 2. Como podemos ver, a diferencia de la cola estática, en lugar de que se recorra nuestro arreglo, se va a recorrer el índice. Esto es porque, como es circular, reutiliza las posiciones vacías, es decir, que, si necesitamos un nuevo elemento, se agregará a los espacios vacíos, como se ve en la figura.

Volvamos a meter un nuevo elemento, que será F, al ser un elemento nuevo y no pasar el límite de la cola, este ocupará el lugar cero como se ve en el inciso c), por el tipo de estructura que usamos. Ya que al tener elementos en la última posición del arreglo (posición 4), se tiene que desplazar de nuevo al inicio del arreglo. Por eso es que tenemos a F en la primera posición, ya que tiene que ocupar los espacios libres, pero no por eso imprimirá primero la F, sino la C, ya que ese es el primer elemento de la fila.

Ahora saquemos otro elemento y metamos dos, creo que ya sabemos que pasará. En el inciso d) de la figura, notamos que se sacamos a C y metimos G y H; de nuevo, el primer elemento de la cola ahora es la D y el último la H, si intentamos meter otro elemento, este no podrá, porque ya está la pila llena, y si queremos hacer un recorrido, entonces se mostrarán los elementos D E F G H, en ese orden.

Tarea/Práctica: implementar por cuenta propia una cola circular, con un máximo de 6 elementos (cola estática).

Cola con prioridad

Una cola con prioridad es similar a una cola normal, solo que este tiene un valor adicional que en su mayoría es un tipo entero, que indica si tiene preferencia o no en la cola, es decir, que en este tipo de cola vamos a tener que ordenar todos los valores, para que el que tenga mayor peso quede hasta el inicio de la cola.

Es similar a las cartas de poker, que una carta tiene mayor valor que otro, veamos en específico las cartas As (A), Rey (K), Reina (Q) y Jota (J). En este ejemplo la prioridad la ordenaremos con el valor mayor, por lo que las cartas quedarán ordenadas como están escritas. Vamos a ver una cola donde la entrada (solo en este caso) será por la derecha y la salida será por la izquierda, entonces, imaginemos que ingresamos a la cola la carta Q, como es el único elemento en la cola no es necesario ordenar. Ahora ingresemos el valor de J, por lo que la cola quedara con [Q J], y si notamos este queda ordenado.

Pero ¿qué sucede si ingresamos al A?, la cola, al hacer la inserción, queda [Q J A] digamos que, por el momento, todo lo que hemos hecho es la ejecución de una cola ordinaria, ahora nos toca hacerla con prioridad. Por el momento, hemos visto el dato núcleo de cada carta, la figura, pero es momento de ver que tanto peso tiene cada una entonces la cola, junto con el valor de prioridad de cada dato, ser vería algo así: [(Q,12) (J,11) (A,14)]. Si observamos detenidamente, el dato A es el valor más alto, y nuestra cola con prioridad (en este caso) debe tener el valor más alto en la salida (lado izquierdo), por lo que debemos usar un método de ordenamiento. Aplicamos nuestra función de ordenar y la cola quedaría de esta forma [A Q J], ahora sí, tenemos nuestra cola ordenada.

Por último, ingresemos a K, entonces nuestra pila quedará con los siguientes elementos [A Q J K], pero vemos que el valor del rey es mayor que al de J y Q, entonces, tendremos que volver a ordenar, y finalmente quedaría así [A K Q J]. Finalmente, tenemos nuestra cola de

prioridad ordenada y si notamos, los valores de prioridad quedan de la siguiente manera [14 13 12 11]. Entonces el primero en salir siempre será el que tenga un peso mayor al de todos.

Como vemos, la principal característica de este tipo de colas es que tiene un dato, y a su vez, tiene un dato de prioridad. Hablando específicamente de un tipo de cola (dinámica), decimos que su nodo tiene tres valores: dato, prioridad, apuntador al nodo siguiente (Figura 2).

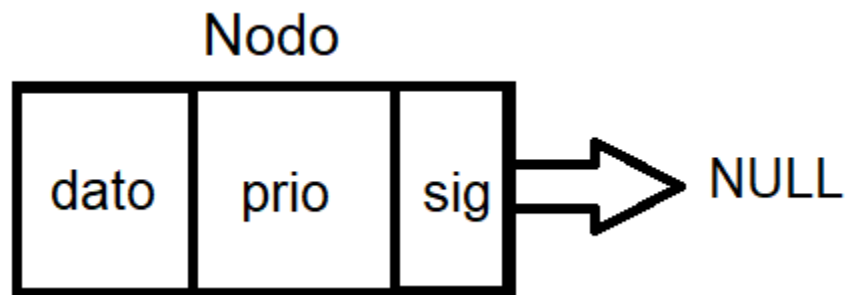


Figura 2. Estructura nodo de una cola con prioridad

Como ya vimos en el ejemplo anterior este tipo de colas necesita de una prioridad que le diga si el dato está por encima de otros, para después, si es que está en el lugar que no le corresponde, ordenarlo. Ya hablando un poco más de programación, este nodo tiene un dato que puede ser de cualquier tipo, una prioridad de preferencia un valor entero y un apuntador siguiente, que se dirige al siguiente nodo, además, se necesita usar de un método de ordenamiento, que siempre va a tomar como referencia de orden la prioridad del dato (Para ver algunos tipos de ordenamiento, véase <https://delta.cs.cinvestav.mx/~adiaz/anadis/Sorting2.pdf>).

Tarea/Práctica: Con ayuda de la cola dinámica, modificar los datos necesarios, para poder hacer una cola con prioridad, con el mejor método de ordenamiento, que usted piense que sea el mejor.