

# Tarea 1

## INFO088 I-2020 - Taller de Estructuras de Datos y Algoritmos.

Profesores: Erick Araya, Héctor Ferrada.  
Instituto de Informática, Universidad Austral de Chile.

Julio 03, 2020

**Entrega.** Debe subir todo en un archivo, t1Alumnos.zip (4 integrantes por grupo), con su implementación (que incluya el Makefile) y un informe claro y detallado a la plataforma siveducmd, Tarea 1, a más tardar el **lunes 27 de Julio de 2020**.

### Implementación de Estructuras y Comparación Empírica.

El objetivo de este trabajo es adquirir un mayor dominio en la implementación de algunas estructuras de datos y algoritmos estudiados en clases, además de utilizar algunas de las funcionalidades de la STL del sistema. Para esto, construirá 3 soluciones de diferentes estructuras tipo queue, codificando además ciertos métodos de ordenamiento, y probará experimentalmente el performance de estas estructuras.

#### Estructuras tipo Queue

El problema consiste en insertar/eliminar datos sobre tres implementaciones diferentes para queue. La idea es que testeé el performance de enqueue/dequeue de cada una de estas estructuras, construidas sobre la misma data, a fin de entregar un buen análisis y las posteriores conclusiones en base a sus experimentos. Las estructuras queue a implementar/testear son las siguientes:

1. **Simple Queue.** Esta es la misma estructura para queue codificada en clases, la cual es construida en base a una lista simplemente enlazada.
2. **Double Queue.** Estructura similar a Simple Queue, salvo que ahora utilizará una lista doblemente enlazada en su implementación. Además, debe tener siempre dos punteros: uno al frente de la cola (llámese  $f$  por ejemplo) y otro al final de la cola (llámese  $r$  por ejemplo). Al tener siempre identificado el primer nodo (con  $f$ ) y el último nodo (con  $r$ ) puede implementar de forma óptima tanto enqueue como dequeue, ya que además puede moverse en una lista doblemente enlazada en cualquier dirección.
3. **Array Queue.** Estructura que implementa una cola en base a un arreglo redimensionable — véase la sección siguiente que explica esta metodología.

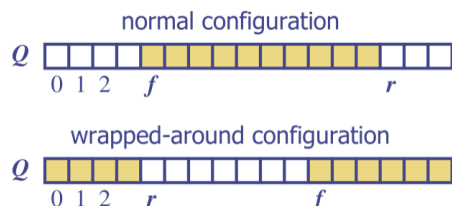
#### Queue en Base a un Arreglo (Array Queue)

En clases se estudió la estructura queue en base a una lista enlazada simple. La idea ahora es reemplazar la lista por un arreglo (llámese  $A[\dots]$ ), que inicialmente tiene una capacidad  $cap$ . Así, cada vez que desee encolar un nuevo dato, tendrá que chequear si aún hay espacio en  $A$ ; de no ser así, creará un nuevo arreglo  $A'$  del doble de la capacidad actual, traspasará todos los datos junto al nuevo elemento en el

nuevo arreglo  $A'$ , el cual reemplaza al anterior (ahora  $A$  será  $A'$  y eliminará el anterior  $A$ ).

El esquema que debe implementar lo ilustra la siguiente figura, donde  $f$  es el frente de la cola (por donde se eliminan los elementos (dequeue)) y  $r$  es la cola (o *rear* en inglés), que es por donde se agregan los elementos (enqueue)). Note que esta estructura es circular.

Una buena opción es que use solo  $cap - 1$  celdas del arreglo y deje la celda de  $r$  siempre vacía, esto le ayudará con las validaciones: si la cola está vacía, entonces  $f == -1$  y  $r == 0$ ; y si la cola está llena, entonces  $r + 1 == f$  (teniendo cuidado de la condición de lista circular).



## Datasets para la Construcción de las Estructuras

Junto a la tarea se les dará el archivo words\_alpha.zip, el cual contiene palabras en inglés. Los elementos a ingresar en las colas deben ser palabras provenientes de este diccionario (seleccionadas aleatoriamente). Utilice el tipo de datos `char*` en las colas para ingresar elementos, es decir arreglos de caracteres. Puede tomar como largos mínimo y máximo a  $MIN$  y  $MAX$  respectivamente, definidos en la cabecera de su programa. Así valida que la palabra a ingresar se encuentre en este rango. Ejemplo  $MIN = 8$  y  $MAX = 24$  es una buena configuración. Las palabras a ingresar en las colas deben ser tomadas aleatoriamente desde el diccionario. Recuerde que, cada vez que haga una inserción, la misma palabra debe ser ingresada en las tres colas, así tendrá tres estructuras con datos idénticos y sus experimentos serán objetivos.

## Experimento enqueue/dequeue

Para cada una de las tres estructuras usted ingresará y eliminará elementos de forma alternada de la siguiente manera:

1. Cree la cola con un solo dato aleatorio.
2. Ingrese y elimine datos alternadamente de acuerdo a la siguiente regla:
  - a) Ingrese (enqueue)  $2k$  palabras
  - b) Elimine (dequeue)  $k/2$  palabras

$k$  es la cantidad de elementos existente en su estructura. Al inicio  $k = 1$  ya que ingresó una sola palabra en el paso 1. Itere de esta forma alternada hasta que su estructura alcance  $n$  elementos, para un  $n$  dado en los argumentos del programa.

3. Una vez completado los  $n$  items, vacíe (dequeue) todos los elementos de la estructura.

Debe probar con diferentes valores significativos para  $n$ ; por ejemplo  $n = \{10^3, 10^4, 10^5, 10^6\}$ .

Puede usted proponer experimentos adicionales para enqueue/dequeue e incluso alguna otra operación adicional si lo estima conveniente. Con sus resultados genere gráficas que ilustren el performance de las estructuras y las operaciones realizadas en sus experimentos; proponga además el tipo de gráfica más adecuada para ilustrar la diferencia entre el rendimiento de cada opción. La idea es que estas medidas le ayuden con las conclusiones de su trabajo. Recuerde que este es un taller de estructuras de datos y algoritmos; por tanto, se espera que concluya respecto a un buen análisis.