Diego Arteaga, Rodrigo Jara, Sebastian Jouannet

Sección 1, Estructuras de datos

**Descripción de algoritmo Transbordador:**

# ***Clases.***

-Nodo:

-Listas: es la clase encargada de crear listas dentro de las cuales se ingresan los datos de los autos y la “hora” de su llegada, dentro de este clase existen los siguientes métodos: Constructor( Orden(1)), add( Orden(n)) y estaVacia( Orden(1)).

-Cola: es la clase encargada de crear colas en las cuales, se guarda el lugar de destino del auto (izquierda o derecha), dentro de esta clase existen los siguientes métodos:

Constructor( Orden(1)), add( Orden(n)), vehiculoss( Orden(1)), peek( Orden(1)), estaVacia( Orden(1)).

-Auto: es la clase que cuenta con los métodos y atributos para crear un objeto “Auto”, dentro de esta clase el único método existente sería el constructor, el cual sirve para crear el Auto y es de Orden (1).

-main: Es la encargada de hacer que los datos pedidos al usuario sean ingresados, esto mediante un menú. Esta es Orden(1) ya que el proceso es hecho dependiendo de la entrada dada por el usuario.

# ***Métodos.***

Nodo.

-Constructor (con next): este método de Orden(1) nos permite crear un nuevo Nodo, el cual al Constructor se le pasan las variables tipo int “nuevoValor” y Nodo “next”.

-Constructor (sin next): este método de Orden(1) nos permite crear un nuevo Nodo, el cual al Constructor se le pasa la variable de tipo int “nuevoValor”.

Listas.

-Constructor: este método de Orden(1) nos permite crear una nueva lista, el cual al Constructor no se le pasa nada, siendo su única modificación el modificar el head a null.

-add: este método de Orden(n) nos permite crear un nuevo nodo y agregarlo al final de la Lista antes creada.

-estaVacia: este método de Orden(1) nos permite saber si una Lista se encuentra vacía o no, en caso de que esté vacía devuelve true y en caso contrario false.

Cola.

-Constructor: este método de Orden(1) nos permite crear una nueva Cola, el cual al Constructor no se le pasa nada, siendo su única modificación el modificar el rear y el front y colocarlos en null.

-add: este método de Orden(n) nos permite crear un nuevo nodo y agregarlo al final de la Cola antes creada.

-vehiculoss: Método que retorna el primer auto y luego pasa el auto en segunda posición a la primera, siempre y cuando la cola no esté vacía, en caso de que esté vacía retorna null. Esta es de Orden(1) ya que solo devuelve un dato y modifica otro.

-peek: Método que devuelve el primer auto, siempre y cuando la cola no esté vacía, en caso de que esté vacía retorna null. Esta es de Orden(1) ya que solo devuelve un dato.

Auto.

-Constructor: este método de Orden(1) nos permite crear un nuevo auto, el cual al Constructor se le pasa la variable tipo int “valor” el cual será el valor de la variable de nombre “horaLlegada”.

Transbordador en general.

-Constructor: este método de Orden(1) nos permite crear un nuevo transbordador, el cual al Constructor se le pasan como variables dos tipo int “Autos” y “tiempo” los cuales sirven de valor a las variables fijas “tamanioAutos” y “tiempoLlegada”.

-agregaAuto: método que guarda el vehículo en la fila que corresponda dependiendo de si llega a la izquierda o a la derecha. Esta es de Orden(1).

-Emulador: método que calcula el tiempo que se demora en llegar cada auto a su lado correspondiente, esto usando tanto las filas como las listas. Este es de orden O(n^2) ya que tiene dos while anidados.