



ALGORITMOS Y ESTRUCTURAS DE DATOS

Pilas LIFO y Colas FIFO

Guillermo Román Díez

`groman@fi.upm.es`

Universidad Politécnica de Madrid

Curso 2015-2016

Pregunta

¿qué es una pila? ¿qué es una cola?

Pregunta

¿qué es una pila? ¿qué es una cola?

Pregunta

¿qué significa LIFO? ¿y FIFO?

Pregunta

¿qué es una pila? ¿qué es una cola?

Pregunta

¿qué significa LIFO? ¿y FIFO?

- ▶ Las **pilas** y las **colas** son TADs fundamentales con innumerables aplicaciones
- ▶ Se usan en multitud de aplicaciones
 - ▶ Pila de llamadas a métodos
 - ▶ Gestión del historial de acciones / navegación
 - ▶ Sistemas Operativos (round robin)
 - ▶ ...



¿QUÉ ES UN LIFO?

- ▶ LIFO \Rightarrow Last In First Out
- ▶ Una Pila, LIFO o Stack es un TAD contenedor que consiste en una secuencia lineal de elementos donde:
 - ▶ El último elemento **apilado (push)** es el primer elemento en ser **desapilado(pop)**
 - ▶ No existe operación de búsqueda
- ▶ No tiene límite de tamaño (teórico. . .)
 - ▶ Os suena **Stack Overflow?**
- ▶ Veremos dos implementaciones usando
 - ▶ **Listas de posiciones**
 - ▶ **Arrays**

- ▶ Interrogadores: `size`, `isEmpty`
- ▶ Inserción: `push`
- ▶ Borrado: `pop`
- ▶ Acceso: `top`
- ▶ Conversión: `toArray`, `toPositionList`
 - ▶ Siguiendo LIFO, el primer elemento del array será el primer elemento desapilado

- ▶ Interrogadores: `size`, `isEmpty`
- ▶ Inserción: `push`
- ▶ Borrado: `pop`
- ▶ Acceso: `top`
- ▶ Conversión: `toArray`, `toPositionList`
 - ▶ Siguiendo LIFO, el primer elemento del array será el primer elemento desapilado

Ejercicio

Implementar el método boolean `balance(String s)`

- ▶ Implementa una pila usando lista de posiciones
- ▶ Sólo tiene la lista como atributo:
 - ▶ El tamaño de la pila es el de la lista
 - ▶ Apilar (`push`) consiste en insertar al principio
 - ▶ Desapilar (`pop`) consiste en borrar el primer elemento
 - ▶ La consulta se realiza mediante la operación `top`
- ▶ Tiene 4 constructores:
 - ▶ (1) Vacío, (2) un array, (3) una lista posiciones o (4) una pila
- ▶ `pop` y `top` lanzan `EmptyStackException` si la pila está vacía
- ▶ `push` no lanza excepción
- ▶ `toString` y `toArray` usan los métodos de la lista
- ▶ `toPositionList` devuelve *una copia* de la lista atributo

- ▶ Implementa una pila usando un array
 - ▶ Establece un tamaño inicial para el array. Cuando se llena crea uno mayor copiando los elementos (*shallow copy*)
- ▶ Tiene 3 atributos: `arr`, `size`, `defaultCapacity`
 - ▶ El tamaño de la pila es el número de elementos apilados (`size`)
 - ▶ Apilar (`push`) consiste en añadir después del último elemento
 - ▶ Desapilar (`pop`) consiste en borrar el último
- ▶ Tiene 4 constructores:
 - ▶ (1) Vacío, (2) un valor para la capacidad, (3) un array, (4) una lista de posiciones, (5) una pila
- ▶ `pop` y `top` lanzan `EmptyStackException` si la pila está vacía
- ▶ `push` comprueba si estamos al máximo de la capacidad y amplía si es necesario
- ▶ `toString` escribe el array en orden inverso
- ▶ `toArray` y `toPositionList` devuelven los elementos uno a uno insertándolos en el array o en la lista



¿QUÉ ES UN FIFO?

- ▶ FIFO \Rightarrow First In First Out
- ▶ Es un TAD contenedor que consiste en una secuencia lineal de elementos donde el primer elemento **encolado (enqueue)** es el primer elemento en ser **desencolado (dequeue)**
- ▶ No existe una operación de búsqueda
- ▶ No tiene límite de tamaño (teórico)
 - ▶ Presenta las mismas limitaciones que la lista
- ▶ Veremos dos implementaciones usando
 - ▶ **Listas de posiciones**
 - ▶ **Arrays**

- ▶ Interrogadores: `size`, `isEmpty`
- ▶ Inserción: `enqueue`
- ▶ Borrado: `dequeue`
- ▶ Acceso: `first`
- ▶ Conversión: `toArray`, `toPositionList`
 - ▶ El primer elemento será el primer elemento desencolado
- ▶ Un ejemplo típico es la gestión de CPU mediante **RoundRobin**

- ▶ Interrogadores: `size`, `isEmpty`
- ▶ Inserción: `enqueue`
- ▶ Borrado: `dequeue`
- ▶ Acceso: `first`
- ▶ Conversión: `toArray`, `toPositionList`
 - ▶ El primer elemento será el primer elemento desencolado
- ▶ Un ejemplo típico es la gestión de CPU mediante

RoundRobin

```
while(!fifo.isEmpty()) {
    Process p = fifo.first();
    fifo.dequeue();
    int status = p.run(20);
    if (status < 0) {
        p.suspend();
        fifo.enqueue(p);
    }
}
```



CLASE FIFOLIST<E>

- ▶ Implementa una cola usando lista de posiciones
- ▶ Sólo tiene la lista como atributo:
 - ▶ El tamaño de la pila es el de la lista
 - ▶ Encolar (enqueue) consiste en insertar al final
 - ▶ Desencolar (dequeue) consiste en borrar el primer elemento
- ▶ Tiene 4 constructores:
 - ▶ (1) Vacío, (2) un array, (3) una lista posiciones o (4) una cola
- ▶ first y dequeue lanzan EmptyFIFOException si la pila está vacía
- ▶ enqueue no lanza excepción
- ▶ toString y toArray usan los método de la lista
- ▶ toPositionList devuelve *una copia* de la lista atributo

- ▶ Implementa una cola usando un array de forma circular
 - ▶ Establece un tamaño inicial para el array. Cuando se llena crea uno mayor copiando los elementos (*shallow copy*)
- ▶ Tiene 5 atributos: `arr`, `size`, `first`, `last` y `defaultCapacity`
 - ▶ El tamaño de la cola es el número de elementos apilados
 - ▶ Encolar (`push`) consiste en insertar al principio
 - ▶ Desapilar (`pop`) consiste en borrar el primer elemento
- ▶ Tiene 5 constructores:
 - ▶ (1) Vacío, (2) un valor para la capacidad, (3) un array, (4) una lista de posiciones, (5) una cola
- ▶ `pop` y `top` lanzan `EmptyFIFOException` si la pila está vacía
- ▶ `push` comprueba si estamos al máximo de la capacidad y amplía si es necesario
- ▶ `toString` escribe el array en orden inverso
- ▶ `toArray` y `toPositionList` devuelven los elementos uno a uno insertándolos en el vector o en la lista

Pregunta

¿por qué usamos un array de forma circular para las colas y no lo hemos hecho así con las pilas?

- ▶ La cola está vacía cuando `first` y `last` son iguales
- ▶ Se van insertando elementos mediante `last`
- ▶ El incremento de `first` y `last` se realiza de forma circular

```
last = (last + 1) % arr.length ;
```

- ▶ La cola se llena cuando `first` y `last` vuelven a ser iguales
 - ▶ En este caso tenemos que ampliar el tamaño del array

Pregunta

¿puede `last` ser menor que `first`?



AMPLIACIÓN DEL TAMAÑO DEL ARRAY

- ▶ Se puede producir cuando se ejecuta enqueue
- ▶ La ampliación el tamaño de la pila es igual que el tamaño del array
- ▶ Por decisión de diseño **doblamos** el tamaño del array
- ▶ Se crea un array, se copian los elementos del antiguo en la parte inicial del array
 - ▶ La copia hay que hacerla en el orden de encolado
- ▶ `first` pasa a ser el primer elemento del array
- ▶ `last` pasa a ser el primero de los huecos
- ▶ Se inserta el elemento nuevo en el primer hueco