Rotación en una cola S32503_es

Disponemos de una clase `Queue` (fichero `queue.hh`) que implementa una cola de valores de tipo `T`. Se trata de añadir un método `rotate` a la clase `Queue`, que recibe un valor v, y tiene el siguiente comportamiento.

- Si v no se encuentra en la cola, ésta no cambia.
- Si v aparece en la cola, el elemento de valor v pasa a ser el primero de la cola, poniendo todos los anteriores a éste al final de la cola, en el mismo orden en que estaban. (Si la cola contiene varios elementos de valor v, se coge el primero.)

Por ejemplo, si la cola es `x y z a b c d`, y v es `a`, entonces la cola se convierte en `a b c d x y z`. La cabecera del método a implementar es:

```
/**
 * @pre:  Cierto.
 *
 * @post: Si la cola (p. i.) no contiene el valor value, ésta no cambia.
 *        Si la cola lo contiene, se tomará como referencia la primera
 *        aparición, y se moverán todos los elementos anteriores a ésta al
 *        final de la cola, en el mismo orden en que estaban.
 */
void rotate(T value);
```

(El método `rotate` se encuentra al final del fichero `queue.hh`.)

Los ficheros públicos (icono del gatito) incluyen un `.tar` con `main.cc`, `queue.hh` y un `Makefile`. También se incluye una copia de los juegos de prueba públicos por comodidad. En la carpeta donde se descompriman se puede: compilar con "`make`"; y testear con "`make test`".

El `main.cc` ya se encarga de leer la entrada, procesar los comandos y producir la salida. Para entregar, solo es necesario subir al Jutge vuestro archivo `queue.hh` modificado.

Entrada

La entrada del programa es una secuencia de comandos como los siguientes que se irán aplicando sobre una cola de `string`s que está inicialmente vacía:

```
push x     // lee un string y lo pone en la cola
pop
front
size
rotate     // lee un string y rota la cola
print      // muestra la cola a la salida
```

Se puede suponer que la secuencia de entrada será correcta (p.e., sin comandos `pop` ni `front` cuando la cola está vacía).

Salida

La salida es el resultado de los comandos `size`, `front` o `print`. Consultad los juegos de prueba, ya que son auto-explicativos.

Observación

La **eficiencia** del algoritmo es importante.

Ejemplo de entrada 1

```
push x
rotate a
size
push a
print
rotate a
print
```

Ejemplo de salida 1

```
1
2 x a
2 a x
```

Ejemplo de entrada 2

```
push a
push b
push c
rotate b
print
push d
print
pop
rotate d
print
push c
print
rotate c
print
```

Ejemplo de salida 2

```
3 b c a
4 b c a d
3 d c a
4 d c a c
4 c a c d
```

Información del problema

Autor : PRO2
Generación : 2025-01-09 09:28:52