
Ordenació eficient d'una cua circular**X89588_ca**

Donada la classe *cua* que permet encuar enters amb una estructura simplement encadenada i circular, cal implementar el mètode

void ordena()

que ordena eficientment els elements del paràmetre implícit de menor a major. No es poden usar estructures de dades auxiliars com els vectors o arrays.

Cal enviar a jutge.org la següent especificació de la classe *cua* i la implementació del mètode dins del mateix fitxer. Al principi de cada mètode implementat, dins d'un comentari, cal indicar el cost temporal amb el raonament corresponent, incloent l'equació de la recurrència si fos necessari.

#include <iostream>

#include <sstream>

using namespace *std*;

typedef unsigned int *nat*;

class *cua* {

 // cua simplement encadenada, sense fantasma i circular.

public:

cua();

 // Pre: True

 // Post: El p.i. és una cua buida.

 ~*cua*();

 // Pre: True

 // Post: El p.i. s'ha destruït.

void encuar(**const int** &*x*);

 // Pre: True

 // Post: Afegeix l'element *x* al final de la cua.

void desencuar();

 // Pre: True

 // Post: Treu el primer element de la cua. Llança un error si la cua és buida.

const int& primer() **const**;

 // Pre: True

 // Post: Obté el primer element de la cua. Llança un error si la cua és buida.

bool es_buida() **const**;

 // Pre: True

 // Post: Consulta si la cua és buida o no.

nat longitud() **const**;

 // Pre: True

 // Post: Retorna el nombre d'elements del p.i.

```

void ordena();
// Pre: True
// Post: S'han ordenat eficientment els elements del p.i. de menor a major

static const int CuaBuida = 310;

private:
    struct node {
        int info; // Informació del node
        node *seg; // Punter al següent element
    };
    node *_ult; // Punter a l'últim element
    nat _long; // Nombre d'elements

    // Aquí va l'especificació dels mètodes privats addicionals
};

// Aquí va la implementació del mètode ordena i dels privats addicionals

```

Per testejar la solució, jutge.org ja té implementats la resta de mètodes de la classe *cua* i un programa principal que processa línies d'enters amb els que crea cues i després crida el mètode *ordena*.

Entrada

L'entrada conté diverses línies formades per seqüències d'enters. Cadascuna d'elles són els elements que tindrà cada cua.

Sortida

Per a cada línia d'entrada, escriu una línia amb el resultat després d'haver ordenat els elements: El nombre d'elements de la cua seguit d'un espai, els elements de la cua entre claudàtors i separats per espais.

Observació

Només cal enviar la classe requerida i la implementació del mètode *ordena*. Pots ampliar la classe amb mètodes privats. Segueix estrictament la definició de la classe de l'enunciat. No es poden usar estructures de dades auxiliars com els vectors o arrays.

Al principi de cada mètode implementat i dins d'un comentari cal indicar el cost temporal amb el raonament corresponent, incloent l'equació de la recurrència si fos necessari.

Exemple d'entrada 1

Exemple d'entrada 2

2

Exemple de sortida 1

0 []

Exemple de sortida 2

1 [2]

Exemple d'entrada 3

-2

Exemple d'entrada 4

2 5 1 9

Exemple d'entrada 5

-2 5 2 4

Exemple d'entrada 6

1 2 3 4 5 6 7 8 9
9 8 7 6 5 4 3 2 1 0

Exemple de sortida 3

1 [-2]

Exemple de sortida 4

4 [1 2 5 9]

Exemple de sortida 5

4 [-2 2 4 5]

Exemple de sortida 6

9 [1 2 3 4 5 6 7 8 9]
10 [0 1 2 3 4 5 6 7 8 9]

Informació del problema

Autoria: Jordi Esteve

Generació: 2026-01-25T17:06:34.410Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>