
HotSet**Y89872_ca**

En aquest exercici implementareu una nova classe de dades, anomenada `HotSet`. Els objectes d'aquesta classe representen conjunts d'elements d'un cert tipus `T` sense repeticions, ordenats per ordre de consulta o d'inserció. La idea d'aquesta estructura és que els elements més freqüentment consultats o inserits quedin a l'inici de l'estructura, i per tant es redueixi molt el cost d'accedir-hi, ja que la cerca és lineal.

El cas d'ús d'un `HotSet` són els problemes on la probabilitat de tornar a utilitzar un element del `HotSet` que s'ha usat recentment és molt més alta que la probabilitat d'usar altres elements.

La classe s'implementa amb una llista simplement encadenada: els elements tenen l'adreça del següent, però no de l'anterior. L'estructura també té un sentinella, de nom `sent`, que és un element que marca l'inici, però no té contingut.

La plantilla del fitxer `hot_set.hh` per fer la implementació és la següent:

```
#ifndef HOT_SET_HH
#define HOT_SET_HH

#include <iostream>
#include <string>
using namespace std;

template <typename T>
class HotSet {
public:
    HotSet() {
        tam = 0;
        sent = new nodo;
        sent->seg = nullptr;
    }

    /// >>>> Implementar aquí els mètodes públics <<<<

private:
    struct nodo {
        T info;
        nodo *seg; // apuntador al següent element
    };

    nodo *sent; // apuntador al sentinella, que *no* és pròpiament un element
    int tam;

    /// >>>> Implementar els mètodes privats necessaris <<<<
};

#endif // HOT_SET_HH
```

Així doncs:

- un `HotSet` que està buit sempre tindrà el sentinella, i es compleix que `sent->seg == nullptr`;
- si un `HotSet` té un o més elements, `sent->seg` apunta al primer element; i,
- l'últim element d'un `HotSet` no té successors, és a dir, si `ult` és un apuntador a l'últim element, llavors `ult->seg == nullptr`.

Es tracta, doncs, d'implementar els mètodes que falten de la classe `HotSet`, de tal manera que el programa principal (sense modificar) funcioni correctament.

Entrada

El programa principal ja es proporciona (també el `Makefile`), a la icona del gatet, i llegeix una seqüència de comandes que exerceixen les operacions de la classe `HotSet`. Cada comanda és una línia que comença amb una paraula i crida algun mètode del `HotSet`.

Sortida

La sortida també està implementada en el programa principal. Cada vegada que es crida a un mètode de la classe `HotSet`, es mostra algun missatge a la sortida estàndard.

Observació

Envieu un `.tar` tal com el que us podeu baixar a la icona del gatet, afegint el fitxer `hot_set.hh` amb la vostra implementació.

Exemple d'entrada 1

```
ins 1
ins 2
front
ins 3
has 1
front
size
has 3
show
```

Exemple d'entrada 2

```
ins 1
ins 2
ins 3
show
ins 2
show
has 5
has 3
front
ins 1
show
```

Exemple de sortida 1

```
primero: 2
1 pertenece
primero: 1
3 elementos
3 pertenece
3 1 2
```

Exemple de sortida 2

```
3 2 1
2 3 1
5 no pertenece
3 pertenece
primero: 3
1 3 2
```

Informació del problema

Autoria: PRO2

Generació: 2026-01-25T21:39:16.634Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>