
Trie TST. Compta les claus entre dues claus donades. X24161_ca

Donada la classe *dicc* que permet gestionar diccionaris on només hi guardem claus úniques usant tries implementats amb la tècnica d'arbres ternaris de cerca (TST), cal implementar el mètode

```
nat quantes_interval (string inicial , string final ) const;
// Pre: Les claus inicial i final estan en el diccionari
// Post: Retorna el nombre de claus que compleixen: inicial <= clau <= final
```

Les claus són del tipus string i els símbols utilitzats per construir el trie són els chars de les claus. S'ha usat el char especial '#' per indicar la fi de la clau. Els símbols dels nodes germans estan ordenats de menor a major.

Cal enviar a jutge.org la següent especificació de la classe *dicc* i la implementació del mètode dins del mateix fitxer. La resta de mètodes públics i privats ja estan implementats.

```
#include <iostream>
using namespace std;
typedef unsigned int nat;

class dicc {
public:
    // Constructora per defecte. Crea un diccionari buit.
    dicc ();

    // Destructora
    ~dicc ();

    // Insereix la clau k en el diccionari. Si ja hi era, no fa res.
    void insereix (const string &k);

    nat quantes_interval (string inicial , string final ) const;
    // Pre: Les claus inicial i final estan en el diccionari
    // Post: Retorna el nombre de claus que compleixen: inicial <= clau <= final

private:
    struct node {
        char _c; // Símbol posició i-èssima de la clau
        node* _esq; // Fill esquerra, apunta a símbols mateixa posició formant un BST
        node* _cen; // Fill central, apunta a símbols següent posició
        node* _dre; // Fill dret, apunta a símbols mateixa posició formant un BST
        node(const char &c, node* esq = NULL, node* cen = NULL, node* dre = NULL);
    };
    node* _arrel ;

    static void esborra_nodes (node* t);
    static node* insereix (node *t, nat i, const string &k);
```

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

```
// Aquí va la implementació del mètode públic quantes_interval i privats addicionals
```

Degut a que jutge.org només permet l'enviament d'un fitxer amb la solució del problema, en el mateix fitxer hi ha d'haver l'especificació de la classe i la implementació del mètode *quantas_interval* (el que normalment estarien separats en els fitxers *.hpp* i *.cpp*).

Per testejar la classe disposes d'un programa principal que insereix claus en un diccionari i després compta quantes hi ha en diferents intervals.

Entrada

L'entrada conté dos blocs separats per una línia amb 10 guions (———). El primer bloc consisteix en una llista de strings: són les claus que tindrà el diccionari. El segon bloc consisteix en una llista de parelles de strings: Són els string inicial i final amb els que comptarem les claus que estan entremig.

Sortida

Per a cada parella de strings d'entrada del segon bloc, escriu una línia amb el nombre de claus que estan entre els dos strings, el text " claus entre " i després els dos string separats pel text " i ".

Observació

Només cal enviar la classe requerida i la implementació del mètode *quantas_interval*. Pots ampliar la classe amb mètodes privats. Segueix estrictament la definició de la classe de l'enunciat.

Per superar els jocs de prova privats, el mètode *quantas_interval* ha de visitar només els nodes del trie imprescindibles.

Pots utilitzar els mètodes i operadors de la classe string, per exemple:

- Operadors comparació: ==, !=, <, >, <=, >=
- Operador concatenació de dos strings o d'un string i un char: +
- Mètode per obtenir la longitud d'un string: length()
- Mètode per obtenir un troç d'un string (les posicions comencen per 0): substr(posicio_1er_caràcter, nombre_de_caràcters).

Exemple d'entrada 1

```
OCA  
-----  
OCA OCA
```

Exemple d'entrada 2

```
CASA  
CAS  
-----
```

Exemple de sortida 1

```
1 claus entre OCA i OCA
```

```
CAS CASA  
CAS CAS  
CASA CASA  
CASA CAS
```

Exemple de sortida 2

2 claus entre CAS i CASA

Exemple d'entrada 3

DAU
DIT
AU
AVI
CASA
COP
CAP
OU
OLA
UN
EXTRA
FUM
FOC
ILLA
ALA

ALA UN
ALA ILLA
CAP CAP
CAP COP
COP DAU
ALA AU
EXTRA FUM
UN UN
UN ALA
COP CAP

Exemple d'entrada 4

DAU
DIT
AU
AVI
CASA
COP
CAP
CAPA
OU
OLA
UN
EXTRA
FUM
FOC
ILLA
ALA
AL

AL UN
AL ILLA
ALA UN
CAP CAPA
CAP CAP
CAP COP
COP DAU
AL AL

1 claus entre CAS i CAS
1 claus entre CASA i CASA
0 claus entre CASA i CAS

Exemple de sortida 3

15 claus entre ALA i UN
12 claus entre ALA i ILLA
1 claus entre CAP i CAP
3 claus entre CAP i COP
2 claus entre COP i DAU
2 claus entre ALA i AU
3 claus entre EXTRA i FUM
1 claus entre UN i UN
0 claus entre UN i ALA
0 claus entre COP i CAP

AL ALA
AL AU
ALA AU
UN UN
UN AL
ALA AL

Exemple de sortida 4

17 claus entre AL i UN
14 claus entre AL i IL·LA
16 claus entre AL·A i UN
2 claus entre CAP i CAP·A
1 claus entre CAP i CAP
4 claus entre CAP i COP

2 claus entre COP i DAU
1 claus entre AL i AL
2 claus entre AL i AL·A
3 claus entre AL i AU
2 claus entre AL·A i AU
1 claus entre UN i UN
0 claus entre UN i AL
0 claus entre AL·A i AL

Informació del problema

Autoria: Jordi Esteve

Generació: 2026-01-25T14:30:47.130Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>