
Ordenació eficient usant un BST**X48149_ca**

Fes un procediment

```
template <typename T>
void ordena(vector <T>& v);
```

que ordeni @v@ de petit a gran amb un algorisme d'ordenació eficient que utilitzi un **BST** per aconseguir-ho. Ha de tenir un cost quasi lineal en el cas mig. El tipus @T@ admet una relació d'ordre total, és a dir, tenim una operació de comparació < entre valors de tipus @T@. Es proporciona una classe *bst* amb els mètodes constructor i destructor ja implementats. Cal enviar a jutge.org la següent especificació de la classe *bst* i la implementació dels mètodes addicionals que creguis convenients dins del mateix fitxer. En el mateix fitxer s'ha d'incloure el procediment *ordena*. Pots ampliar la classe *bst* amb els mètodes públics i privats que necessitis per poder implementar l'ordenació eficient.

```
#include <iostream>
```

```
#include <vector>
```

```
using namespace std;
```

```
typedef unsigned int nat;
```

```
template <typename Clau>
```

```
class bst {
```

```
    public:
```

```
        // Constructora per defecte. Crea un BST buit.
```

```
        bst ();
```

```
        // Destructora
```

```
        ~bst ();
```

```
        // Aquí va l'especificació dels mètodes públics addicionals
```

```
    private:
```

```
        struct node {
```

```
            Clau _k;        // Clau
```

```
            node* _esq;    // fill esquerre
```

```
            node* _dret;   // fill dret
```

```
        };
```

```
        node *_arrel;
```

```
        static void esborra_nodes(node* m);
```

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

```
};
```

```
// Aquí va la implementació dels mètodes públics i privats de bst
```

```
// Aquí va la implementació del procediment ordena
```

En els següents exemples, l'entrada consisteix en diverses línies cadascuna d'elles representant un vector: El nombre d'elements del vector seguit dels seus valors. La sortida mostra els elements de cada vector un cop ordenats.

Exemple d'entrada 1

```
1 10
0
2 10 20
2 20 10
3 10 20 30
3 10 30 20
3 20 30 10
3 20 10 30
3 30 10 20
3 30 20 10
4 1 2 3 4
4 1 2 4 3
4 1 3 2 4
4 1 3 4 2
4 1 4 2 3
4 1 4 3 2
4 4 2 3 1
4 4 2 1 3
4 4 3 2 1
4 4 3 1 2
4 4 1 2 3
4 4 1 3 2
6 4 1 3 2 1 4
6 4 1 3 2 3 3
2 0.0331172488308 0.153664419108
```

Exemple de sortida 1

```
10
10 20
10 20
10 20 30
10 20 30
10 20 30
10 20 30
10 20 30
10 20 30
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
1 2 3 4
1 1 2 3 4 4
1 2 3 3 3 4
0.0331172 0.153664
```

Informació del problema

Autoria: Jordi Esteve

Generació: 2026-01-25T16:30:25.990Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>