

ESTRUCTURA DE DATOS

Práctica Sobre Contenedor

Rafael Sanjuán Aguilera
Juan José Martín Cara
Pablo Torres Rosel

Eficiencia teórica y empírica:

1. Inserción.

```
bool conjunto::insert( const conjunto::value_type & e ) {

    pair<conjunto::value_type,bool> par = find ( e );
    bool existe_ya = par.second;

    if( !existe_ya )
    {
        // Amplio el vector una posición más.
        conjunto::value_type m;
        vm.push_back( m );          //O(1)

        // Busco primer elemento mas grande.
        conjunto::iterator upper = begin(); ///O(1)

        do { upper++; }             //O(n)
        while ( *upper < e );

        // Muevo todos los elementos una posición hacia la derecha
        for ( iterator iter = end()-1; iter != upper ; iter-- ) //O(n)
            *(iter+1) = *(iter);
    }

    return existe_ya;
}
```

Como podemos ver en las anotaciones del código, encontramos dos bucles sin anidar además de unas funciones ya definidas por la stl, al ser este el caso podemos afirmar que la eficiencia teórica es **O(n)**

2. Búsqueda.

```
pair<conjunto::value_type,bool> conjunto::find ( const conjunto::value_type & e ) const
{
    conjunto::value_type m;
    pair<conjunto::value_type,bool> par( m, false );

    for ( unsigned i = 0; i < vm.size() && !par.second; i++ ) //O(n)
    {
        if( vm[ i ].getID() == e.getID() )
        {
            par.first = vm[ i ];
            par.second = true;    //O(1)
        }
    }

    return par;
}
```

Tenemos un único “for” con lo cual la eficiencia teórica queda reducida a esta misma eficiencia **O(n)**

3. Borrado.

```
bool conjunto::erase( const conjunto::value_type & e ) {  
    bool deleted = false;  
    for ( unsigned i = 0; i < vm.size(); i++ ) //O(n)  
    {  
        if( vm[ i ].getID() == e.getID() )  
        {  
            vm.erase( vm.begin() + i ); //O(n)  
            deleted = true;  
        }  
    }  
    return deleted;  
}
```

En este caso la eficiencia es la misma que las dos anteriores teniendo por tanto una eficiencia teórica de **O(n)** dado que el erase de la stl solo es llamado una única vez.