

priority_queue<T, vector<T>,Compare>

Categorías: contenedores

Tipo componente: tipo

Descripción

Una cola_con_prioridad (`priority_queue`) es un contenedor que proporciona un subconjunto restringido de métodos (inserción de elementos además del borrado y consulta del elemento “superior”). Se garantiza que el elemento “superior” de la `priority_queue`, es elemento más grande, donde el objeto de la función `compara` se utiliza para las comparaciones. `Priority_queue` no permite la iteración a través de sus elementos.

Ejemplo

```
int main() {  
  
    priority_queue<int> Q;  
  
    priority_queue<int,vector<int>, greater<int> > Q2;  
  
    Q.push(1);  
    Q.push(4);  
    Q.push(2);  
    Q.push(8);  
    Q.push(5);  
    Q.push(7);  
  
    assert(Q.size() == 6);  
  
    assert(Q.top() == 8);  
    Q.pop();  
  
    assert(Q.top() == 7);  
    Q.pop();  
  
    assert(Q.top() == 5);  
    Q.pop();  
  
    assert(Q.top() == 4);  
    Q.pop();  
  
    assert(Q.top() == 2);  
    Q.pop();  
  
    assert(Q.top() == 1);  
    Q.pop();  
  
    assert(Q.empty());  
}
```

Definición

Definido en el fichero `cabecera <queue>` , y en el fichero `<queue.h>` para compatibilidad con versiones anteriores

Parámetros de la plantilla

Parámetro	Descripción	Defecto
T	El tipo de objeto almacenado en la cola con prioridad.	
Vector	El contenedor sobre el que se implementa.	Vector<T>
Comparar	La función de la comparación que determinaba si un elemento es más pequeño que otro elemento. Si <code>comparar (x, y)</code> es verdad, entonces <code>x</code> es más pequeño que el <code>Y</code> . El elemento vuelto por <code>Q.top ()</code> es el elemento más grande de la cola con prioridad. Es decir, tiene la característica que, para cada otro elemento <code>x</code> en la cola con prioridad, <code>compara (Q.top (), x)</code> es falso.	less<T>

Requerimientos del tipo.

- T es un tipo que tiene el operador de asignación.
- Comparar es un objeto de comparación
- Comparar induce un orden sobre los objetos del tipo T
- Los objetos del tipo T se pueden comparar mediante el objeto Comparar

Miembros

Miembro	Descripción
<code>size_type</code>	Unsigned int
<code>priority_queue ()</code>	El constructor por defecto. Crea un <code>priority_queue</code> vacío, usando <code>less<T></code> como función de comparación.
<code>priority_queue (const priority_queue &)</code>	El constructor de copia.

priority_queue & operator=(const priority_queue &)	El operador de asignación.
bool empty() const	Devuelve verdad si el priority_queue no contiene ningún elemento, y falso de otra manera. S.empty () es equivalente a S.size ()==0 .
size_type size() const	Devuelve el número de los elementos contenidos en el priority_queue .
const T & top() const	Devuelve una referencia constante al elemento en el tope del priority_queue . Precondición: vacío () es falso Se garantiza que el elemento en el tope es el elemento más grande de la cola con prioridad, según determina la función de la comparación comparar . Es decir, para cada otro elemento x en el priority_queue , comparar (Q.top (), x) es falso.
void push(const T & x)	Insertamos x en el priority_queue . Postcondición: el tamaño () será incrementado en 1 .
Void pop()	Quita el elemento en el tope del priority_queue , es decir, el elemento más grande Precondición: vacío () es falso Postcondición: el tamaño () será decrementado en 1 .