Bài 5.7: Thư viện priority_queue

- ✓ Đặc điểm
- ✓ Các hàm thông dụng và mô tả
- ✓ Ví dụ minh họa
- ✓ Bài tập thực hành

Đặc điểm

- Một hàng đợi ưu tiên là một container adaptor cho phép truy cập vào phần tử có mức ưu tiên lớn nhất với thời gian hằng số.
- Ngoài ra hàng đợi ưu tiên cũng hỗ trợ mở rộng thao tác chèn và xóa dữ liệu với thời gian logarit.
- Để sử dụng priority_queue ta include thư viện <queue>.
- Mặc định priority_queue sắp xếp các phần tử theo thứ tự từ lớn đến nhỏ.
- Ta có thể sử dụng std::greater<T> để thay đổi cách sắp đặt thứ tự ưu tiên cho các phần tử trong hàng đợi.
- Tương tác với priority_queue tương tự như quản lý heap trong container truy cập ngẫu nhiên.

Các hàm thông dụng và mô tả

```
priority_queue() : priority_queue(Compare(), Container()) { }
explicit priority_queue( const Compare& compare )
: priority_queue(compare, Container()) { }
explicit priority_queue( const Compare& compare = Compare(),
                         const Container& cont = Container() );
priority_queue( const Compare& compare, const Container& cont );
priority_queue( const Compare& compare, Container&& cont );
priority_queue( const priority_queue& other );
priority_queue( priority_queue&& other );
template< class InputIt > priority_queue( InputIt first, InputIt last,
const Compare& compare = Compare() );
template< class InputIt >
priority_queue( InputIt first, InputIt last,
                const Compare& compare = Compare(),
                const Container& cont = Container() );
template< class InputIt >
priority_queue( InputIt first, InputIt last,
                const Compare& compare, const Container& cont );
template< class InputIt >
priority_queue( InputIt first, InputIt last,
                const Compare& compare, Container&& cont );
```

Các hàm thông dụng và mô tả

priority_queue& operator=(const priority_queue& other); hoặc priority_queue& operator=(const priority_queue&& other) – Thay thế nội dung của hàng đợi hiện tại bằng nội dung của hàng đợi khác.
const_reference top() const – Trả về tham chiếu hằng đến phần tử top trong hàng đợi ưu tiên. Phần tử top là phần tử có thứ tự ưu tiên cao nhất trong hàng đợi ưu tiên. Kết quả trả về của hàm này tương tự gọi hàm front() của container.
void pop() – Xóa phần tử top khỏi hàng đợi ưu tiên. Hành động này giảm kích thước của hàng đợi đi 1. Phần tử bị xóa là phần tử có mức ưu tiên cao nhất trong hàng đợi.
void push(const value_type& val) – Thêm phần tử mới vào hàng đợi ưu tiên.
bool empty() const – kiểm tra xem liệu hàng đợi có đang rỗng không. Trả về true nếu hàng đợi rỗng và false trong trường hợp ngược lại.
size_type size() const – Trả về kích thước hiện tại của hàng đợi. Tức số phần tử hiện có của hàng đợi.
template<class... Args> void emplace(Args&&... args) – Tạo và chèn phần tử mới vào hàng đợi. Phần tử mới được tạo tại chỗ.
void swap(priority_queue& x) noexcept() – Tráo đổi các phần tử trong hai hàng đợi.

Các hàm thông dụng và mô tả

```cpp
void showElements(priority_queue<int> queue) {
    while (!queue.empty()) {
        cout << queue.top() << " ";
        queue.pop();
    }
    cout << endl;
}

int main() {
    priority_queue<int> queue;
    queue.push(100);
    queue.push(500);
    queue.push(300);
    queue.push(700);
    queue.push(200);
    queue.push(600);
    // print result
    cout << "Cac phan tu trong queue mac dinh: \n";
    showElements(queue);
```

Nội dung tiếp theo

Tìm hiểu thư viện <deque>