

Bài 7.9: Một số hàm thao tác heap

- ✓ Tổng quan
- ✓ Các hàm và mô tả
- ✓ Ví dụ minh họa

Tổng quan

- Các hàm thao tác với heap biến đổi các phần tử trong container như vector thành heap.
- Để sử dụng các hàm này ta include thư viện `<algorithm>`.
- Đó là các hàm template có thể tương thích với hầu hết các kiểu dữ liệu được lưu trữ trong tập hợp.
- Khi ta tự định nghĩa các kiểu dữ liệu để sử dụng với các hàm trong thư viện `algorithm`, cần nạp chồng các toán tử của hàm tương ứng. Ví dụ toán tử `<` để so sánh hai phần tử trong container.

Cấu trúc của đồng nhị thức

Tên hàm	Mô tả
<i>bool</i> <i>is_heap(RandomIt first, RandomIt last)</i> (C++11-C++20)	Kiểm tra xem các phần tử trong nửa khoảng [first, last) có phải max heap không. Các phần tử được so sánh bằng toán tử <.
<i>constexpr bool</i> <i>is_heap(RandomIt first, RandomIt last)</i> (C++20)	
<i>bool</i> <i>is_heap(RandomIt first, RandomIt last, Comp comp)</i> (C++11-C++20)	Kiểm tra xem các phần tử trong nửa khoảng [first, last) có phải max heap không. Các phần tử được so sánh bằng hàm so sánh comp.
<i>constexpr bool</i> <i>is_heap(RandomIt first, RandomIt last, Comp comp)</i> (C++20)	
<i>bool</i> <i>is_heap(ExecutionPolicy&& policy, RandomIt first, RandomIt last)</i> (C++17)	Kiểm tra xem các phần tử trong nửa khoảng [first, last) có phải max heap không. Thực hiện theo một policy (quy tắc) cho trước. Các phần tử được so sánh bằng toán tử < hoặc hàm so sánh comp.
<i>bool</i> <i>is_heap(ExecutionPolicy&& policy, RandomIt first, RandomIt last, Comp comp)</i> (C++17)	
<i>void</i> <i>make_heap(RandomIt first, Random last)</i> (Tới C++20)	Tạo heap trong khoảng [first, last). Sử dụng toán tử < để so sánh các phần tử.
<i>constexpr void</i> <i>make_heap(RandomIt first, Random last)</i> (C++20)	
<i>void</i> <i>make_heap(RandomIt first, Random last, Comp comp)</i> (Tới C++20)	Tạo heap trong khoảng [first, last). Sử dụng hàm so sánh comp để so sánh các phần tử.
<i>constexpr void</i> <i>make_heap(RandomIt first, Random last, Comp comp)</i> (C++20)	
<i>void</i> <i>sort_heap(RandomIt first, RandomIt last)</i> (Tới C++20)	Chuyển max heap thành tập hợp được sắp xếp theo thứ tự tăng dần trong khoảng [first, last). Phiên bản này sử dụng toán tử < để so sánh các phần tử.
<i>constexpr void</i> <i>sort_heap(RandomIt first, RandomIt last)</i> (C++20)	

Các thao tác

<code>void sort_heap(RandomIt first, RandomIt last, Comp comp) (Tới C++20)</code>	Chuyển max heap thành tập hợp được sắp xếp theo thứ tự tăng dần trong khoảng [first, last). Phiên bản này sử dụng hàm comp để so sánh các phần tử.
<code>constexpr void sort_heap(RandomIt first, RandomIt last, Comp comp) (C++20)</code>	
<code>void push_heap(RandomIt first, RandomIt last) (Tới C++20)</code>	Thêm phần tử tại vị trí last-1 vào max heap được định nghĩa trong khoảng [first, last - 1). Phiên bản này sử dụng toán tử < để so sánh các phần tử.
<code>constexpr void push_heap(RandomIt first, RandomIt last) (C++20)</code>	
<code>void push_heap(RandomIt first, RandomIt last, Comp comp) (Tới C++20)</code>	Thêm phần tử tại vị trí last-1 vào max heap được định nghĩa trong khoảng [first, last - 1). Phiên bản này sử dụng hàm comp để so sánh các phần tử.
<code>constexpr void push_heap(RandomIt first, RandomIt last, Comp comp) (C++20)</code>	
<code>void pop_heap(RandomIt first, RandomIt last) (Tới C++20)</code>	Loại bỏ phần tử lớn nhất khỏi heap bằng cách tráo đổi nó với phần tử cuối(last-1). Cập nhật heap với các phần tử trong đoạn [first, last-1). Phiên bản này sử dụng toán tử < để so sánh các phần tử.
<code>constexpr void pop_heap(RandomIt first, RandomIt last) (C++20)</code>	
<code>void pop_heap(RandomIt first, RandomIt last, Comp comp) (Tới C++20)</code>	Loại bỏ phần tử lớn nhất khỏi heap bằng cách tráo đổi nó với phần tử cuối(last-1). Cập nhật heap với các phần tử trong đoạn [first, last-1). Phiên bản này sử dụng hàm comp để so sánh các phần tử.
<code>constexpr void pop_heap(RandomIt first, RandomIt last, Comp comp) (C++20)</code>	

Nội dung tiếp theo

Tìm hiểu về bảng băm