堆的STL实现代码

主讲人：邓哲也

堆的STL实现代码

- 在STL中，有一个封装好的优先队列——priority_queue，可以直接当堆使用。

- 使用方法：
- #include <queue>
- priority_queue<int> q;

堆的STL实现代码

- 成员函数：
- q.top()　　获得最大的元素
- q.empty()　　判断是否为空
- q.size()　　返回堆的大小
- q.push(…)　　插入元素
- q.pop()　　删除第一个元素

堆的STL实现代码

定义一个打印堆中元素的函数

```
template<typename T> void print_queue(T& q) {
    while(!q.empty()) {
        std::cout << q.top() << " ";
        q.pop();
    }
    std::cout << '\n';
}
```

堆的STL实现代码

分析这段代码和输出。

```cpp
int main() {
    std::priority_queue<int> q;

    for(int n : {1,8,5,6,3,4,0,9,7,2})
        q.push(n);

    print_queue(q);

    std::priority_queue<int, std::vector<int>, std::greater<int> > q2;

    for(int n : {1,8,5,6,3,4,0,9,7,2})
        q2.push(n);

    print_queue(q2);

    // 用 lambda 比较元素。
    auto cmp = [](int left, int right) { return (left ^ 1) < (right ^ 1);};
    std::priority_queue<int, std::vector<int>, decltype(cmp)> q3(cmp);

    for(int n : {1,8,5,6,3,4,0,9,7,2})
        q3.push(n);

    print_queue(q3);

}
```

输出：

```
9 8 7 6 5 4 3 2 1 0
0 1 2 3 4 5 6 7 8 9
8 9 6 7 4 5 2 3 0 1
```

下节课再见