

Maximum N elements

Pieter P

C++ 20

C++ 20 Ranges Implementation

```
1 #include <algorithm> // ranges::partial_sort_copy
2 #include <ranges>    // views::iota, ranges::forward_range, ranges::borrowed_iterator_t
3 #include <vector>    // vector
4
5 /**
6  * @brief   Return a vector of iterators to the n largest elements of the given
7  *          range.
8  *
9  * @tparam R The type of range.
10 * @param range The input range to find the largest elements in.
11 * @param n The number of largest elements to return.
12 * @return Vector containing iterators to the n largest elements of the range.
13 */
14 template <std::ranges::forward_range R>
15 constexpr std::vector<std::ranges::borrowed_iterator_t<R>>
16 max_n_elements(R &&range, size_t n) {
17     // The iterator type of the input range
18     using iter_t = std::ranges::borrowed_iterator_t<R>;
19     // The range of iterators over the input range
20     auto iterators = std::views::iota(std::ranges::begin(range),
21                                     std::ranges::end(range));
22     // Vector of iterators to the largest n elements
23     std::vector<iter_t> result(n);
24     // Lambda function to compare two iterators: dereference them and compare
25     // their values
26     auto compare = [](iter_t it_a, iter_t it_b) { return *it_a > *it_b; };
27     // Sort the largest n elements of the input range, and store iterators to
28     // these elements to the result vector
29     std::ranges::partial_sort_copy(iterators, result, compare);
30     return result;
31 }
32
33
34
35
```

C++ 20 Driver

```
37 #include <iostream>
38
39 int main() {
40     std::vector<int> v = {10, 12, 8, -5, 8, 3, -2, 1, 9, 12};
41     size_t n = 4;
42     auto max_elems = max_n_elements(v, n);
43     std::cout << "Largest elements:" << std::endl;
44     for (auto it : max_elems) {
45         auto idx = it - v.begin();
46         std::cout << "    [" << idx << "]:\t" << *it << std::endl;
47     }
48 }
```

```
Largest elements:
[1]: 12
[9]: 12
[0]: 10
[8]: 9
```

C++ 11 Algorithms Implementation

```

1  #include <algorithm> // partial_sort_copy
2  #include <iterator> // forward_iterator_tag
3  #include <vector> // vector
4
5  /// Range iterator type, to iterate over ranges of iterators
6  template <class Iterator>
7  struct RangeIterator {
8      using value_type = Iterator;
9      using reference = const Iterator &;
10     using pointer = void;
11     using difference_type = ptrdiff_t;
12     using iterator_category = std::forward_iterator_tag;
13
14     value_type it;
15
16     RangeIterator(Iterator it) : it(it) {}
17     reference operator*() const { return it; };
18     RangeIterator &operator++() { return ++it, *this; }
19     RangeIterator operator++(int) { RangeIterator t = *this; ++it; return t; }
20     bool operator!=(RangeIterator other) const { return it != other.it; }
21     bool operator==(RangeIterator other) const { return it == other.it; }
22 };
23
24 /**
25  * @brief Return a vector of iterators to the n largest elements of the given
26  * range.
27  *
28  * @tparam InputIt
29  * The type of iterator over the input range.
30  * @param first
31  * The iterator to the beginning of the input range.
32  * @param last
33  * The iterator to the end of the input range.
34  * @param n
35  * The number of largest elements to return.
36  *
37  * @return Vector containing iterators to the n largest elements of the range.
38  */
39 template <class InputIt>
40 constexpr std::vector<InputIt>
41 max_n_elements(InputIt first, InputIt last, size_t n) {
42     // An iterator over the iterators over the input range
43     using iter_iter_t = RangeIterator<InputIt>;
44     // Vector of iterators to the largest n elements
45     std::vector<InputIt> result(n);
46     // Lambda function to compare two iterators: dereference them and compare
47     // their values
48     auto compare = [](InputIt it_a, InputIt it_b) { return *it_a > *it_b; };
49     // Sort the largest n elements of the input range, and store iterators to
50     // these elements to the result vector
51     std::partial_sort_copy(iter_iter_t(first), iter_iter_t(last),
52                           std::begin(result), std::end(result),
53                           compare);
54     return result;
55 }

```

C++ 11 Driver

```

57 #include <iostream>
58
59 int main() {
60     std::vector<int> v = {10, 12, 8, -5, 8, 3, -2, 1, 9, 12};
61     size_t n = 4;
62     auto max_elems = max_n_elements(v.begin(), v.end(), n);
63     std::cout << "Largest elements:" << std::endl;
64     for (auto it : max_elems) {
65         auto idx = it - v.begin();
66         std::cout << " [" << idx << "]:\t" << *it << std::endl;
67     }
68 }

```

Largest elements:

```

[1]: 12
[9]: 12
[0]: 10
[8]: 9

```