What algorithm is this?

```
std::vector<int> v{2, 4, 2, 0, 5, 10, 7};

for (auto i = v.begin(); i != v.end(); ++i)
{
 std::rotate(
  std::upper_bound(v.begin(), i, *i),
  i, i+1);
}
```

C++

For a `std::deque<>`, what is the **time** complexity for **random access**?

C++

For a `std::deque<>`, what is the **time** complexity for random **insertion**?

C++

For a `std::deque<>`, what is the **time** complexity for **popping** from either end?

C++

For a `std::deque<>`, what is the **time** complexity for **pushing** from either end?

C++

For a `std::deque<>`, what is the **time** complexity for random **removal**?

C++

What does a `std::deque<>` container do?

C++

What C++ **container** is similar to a `std::queue<>` but can be pushed and popped from **both ends**?

C++

What is a `std::forward_list`?

C++

What header is STL container `std::forward_list` declared in?

C++

O(1)

AKA: Constant

Insertion sort

O(1)

AKA: Constant

O(N)

AKA: Linear

O(N)

AKA: Linear

O(1)

AKA: Constant

`std::deque<>`

(Pronounced as "deck")

It is similar to a
`std::queue<>`
but can be pushed
and popped from both end.

`forward_list`

Example:

```
#include <forward_list>
```

A C++
template library
implementation
of a **single-linked list**.

What header is function `std::abs` declared in?

Example:

```C++
float f = std::abs(25.5f);
```

What header is function `std::accumulate` declared in?

```C++
```

What header is function `std::adjacent_difference` declared in?

```C++
```

What header is `std::boolalpha` and `std::noboolalpha` declared in?

Example:

```C++
std::cout << std::boolalpha;
std::cout << std::noboolalpha;
```

What header is function `std::ceil` declared in?

Example:

```C++
float f = std::ceil(25.5f);
```

What header is function `std::clamp` declared in?

Example:

```C++
int x = std::clamp<int>(4, 5, 6)
```

What header is function `std::cos` declared in?

Example:

```C++
float f = std::cos(1.0707);
```

What header is `std::defaultfloat` declared in?

Example:

```C++
std::cout << std::defaultfloat;
```

What header is class `std::deque` declared in?

```C++
```

What header is `std::endl` declared in?

Example:

```C++
std::cout << std::endl;
```

numeric

Example:

```
#include <numeric>
```

cmath

Example:

```
#include <cmath>
```

ios

Example:

```
#include <ios>
```

numeric

Example:

```
#include <numeric>
```

algorithm

Example:

```
#include <algorithm>
```

cmath

Example:

```
#include <cmath>
```

ios

Example:

```
#include <ios>
```

cmath

Example:

```
#include <cmath>
```

ostream

Example:

```
#include <ostream>
```

deque

Example:

```
#include <deque>
```

What header is `std::ends` declared in?

Example:

```C++
std::ostringstream oss;
oss << "hello!" << std::ends;
```

What header is `std::find` declared in?

Example:

```C++
std::vector<int> vec = {0, 5, 10};
auto it = std::find( vec, 5);
```

What header is `std::fixed` declared in?

Example:

```C++
std::cout << std::fixed;
```

What header is function `std::floor` declared in?

Example:

```C++
float f = std::floor(25.5f);
```

What header is `std::flush` declared in?

Example:

```C++
std::cout << "hi!" << std::flush;
```

What header is function `std::fmod` declared in?

Example:

```C++
float f = std::fmod(10.2, 1.11);
```

What header is function `std::format` declared in?

C++

What header is function `std::fpclassify` declared in?

C++

What header is `std::get_money` declared in?

C++

What header is `std::get_time` declared in?

C++

algorithm

Example:

```
#include <algorithm>
```

ostream

Example:

```
#include <ostream>
```

cmath

Example:

```
#include <cmath>
```

iomanip

Example:

```
#include <iomanip>
```

cmath

Example:

```
#include <cmath>
```

ostream

Example:

```
#include <ostream>
```

cmath

Example:

```
#include <cmath>
```

format

Example:

```
#include <format>
```

iomanip

Example:

```
#include <iomanip>
```

iomanip

Example:

```
#include <iomanip>
```

What header is `std::hexfloat` declared in?

Example:

```C++
std::cout << std::hexfloat;
```

What header is class `std::initializer_list` declared in?

```C++
```

What header is function `std::inner_product` declared in?

```C++
```

What header are
- std::left
- std::right
- std::internal

declared in?

```C++
```

What header are
- `std::dec`
- `std::hex`
- `std::oct`

declared in?

```C++
```

What header is function `std::is_heap_until` declared in?

```C++
```

What header is function `std::lcm` declared in?

```C++
```

What header is function `std::lerp` declared in?

```C++
```

What header is `std::make_heap` declared in?

Example:

```C++
std::vector<int> vec = {0, 5, 10};
std::make_heap(vec.begin(), vec.end())
```

What header is function `std::max_element` declared in?

```C++
```

initializer_list

Example:

```
#include <initializer_list>
```

ios

Example:

```
#include <ios>
```

ios

Example:

```
#include <ios>
```

numeric

Example:

```
#include <numeric>
```

algorithm

Example:

```
#include <algorithm>
```

ios

Example:

```
#include <ios>
```

cmath

Example:

```
#include <cmath>
```

numeric

Example:

```
#include <numeric>
```

algorithm

Example:

```
#include <algorithm>
```

algorithm

Example:

```
#include <algorithm>
```

What header is function
`std::max`
declared in?

Example:

```C++
int x = std::max<int>(4, 5)
```

What header is function
`std::midpoint`
declared in?

What header is function
`std::min_element`
declared in?

What header is function
`std::min`
declared in?

Example:

```C++
int x = std::min<int>(4, 5)
```

What header is function
`std::minmax`
declared in?

What header is class
`std::multimap`
declared in?

What header is class
`std::multiset`
declared in?

What header is
template class
`std::numeric_limits<>`
declared in?

What header is class
`std::ostringstream`
declared in?

Example:

```C++
std::ostringstream oss;
```

What header is function
`std::pop_heap`
declared in?

numeric

Example:

```
#include <numeric>
```

algorithm

Example:

```
#include <algorithm>
```

algorithm

Example:

```
#include <algorithm>
```

algorithm

Example:

```
#include <algorithm>
```

map

Example:

```
#include <map>
```

algorithm

Example:

```
#include <algorithm>
```

numeric_limits

Example:

```
#include <numeric_limits>
```

set

Example:

```
#include <set>
```

algorithm

Example:

```
#include <algorithm>
```

sstream

Example:

```
#include <sstream>
```

What header is function `std::pow` declared in?

Example:

```C++
float f = std::pow(2.7, 5.0);
```

What header is `std::priority_queue` declared in?

Example:

```C++
std::priority_queue pq;
```

What header is the function `std::push_heap` declared in?

C++

What header is `std::put_money` declared in?

C++

What header is `std::put_time` declared in?

C++

What header is function `std::reduce` declared in?

C++

What header is `std::resetiosflags` declared in?

Example:

```C++
std::cout <<
    std::resetiosflags(
        std::ios_base::basefield);
```

What header is `std::fixed` declared in?

Example:

```C++
std::cout << std::fixed;
```

What header is class `std::set` declared in?

C++

What header is `std::setbase` declared in?

Example:

```C++
std::cout << std::setbase(8);
```

queue

Example:

```
#include <queue>
```

cmath

Example:

```
#include <cmath>
```

iomanip

Example:

```
#include <iomanip>
```

algorithm

Example:

```
#include <algorithm>
```

numeric

Example:

```
#include <numeric>
```

iomanip

Example:

```
#include <iomanip>
```

iomanip

Example:

```
#include <iomanip>
```

iomanip

Example:

```
#include <iomanip>
```

iomanip

Example:

```
#include <iomanip>
```

set

Example:

```
#include <set>
```

What header is `std::setfill` declared in?

Example:

```C++
std::cout << std::setfill('*');
```

What header is `std::setiosflags` declared in?

Example:

```C++
std::cout <<
    std::setiosflags(
        std::ios_base::dec);
```

What header is `std::setprecision` declared in?

Example:

```C++
std::cout << std::setprecision(5);
```

What header is `std::setw` declared in?

Example:

```C++
std::cout << std::setw(12);
```

What header is `std::showpoint` and `std::noshowpoint` declared in?

Example:

```C++
std::cout << std::showpoint;
std::cout << std::noshowpoint;
```

What header is function `std::sin` declared in?

Example:

```C++
float f = std::sin(1.0707);
```

What header is `std::skipws` and `std::noskipws` declared in?

Example:

```C++
std::cout << std::skipws;
std::cout << std::noskipwx;
```

What header is function `std::sort_heap` declared in?

```C++
```

What header is function `std::sort` declared in?

```C++
```

What header is class `std::span` declared in?

```C++
```

iomanip

Example:

```
#include <iomanip>
```

iomanip

Example:

```
#include <iomanip>
```

iomanip

Example:

```
#include <iomanip>
```

iomanip

Example:

```
#include <iomanip>
```

cmath

Example:

```
#include <cmath>
```

ios

Example:

```
#include <ios>
```

algorithm

Example:

```
#include <algorithm>
```

ios

Example:

```
#include <ios>
```

span

Example:

```
#include <span>
```

algorithm

Example:

```
#include <algorithm>
```

What header is function `std::sqrt` declared in?

Example:

```C++
float f = std::sqrt(25.0);
```

What header is function `std::swap` declared in?

```C++
```

What header is function `std::to_string` declared in?

```C++
```

What header is function `std::to_wstring` declared in?

```C++
```

What header is function `std::trunc` declared in?

Example:

```C++
float f = std::trunc(25.5);
```

What header is `std::unitbuf` and `std::nounitbuf` declared in?

Example:

```C++
std::cout << std::unitbuf;
std::cout << std::nounitbuf;
```

What header is function `std::unordered_map` declared in?

```C++
```

What header is function `std::unordered_multimap` declared in?

```C++
```

What header is function `std::unordered_multiset` declared in?

```C++
```

What header is function `std::unordered_set` declared in?

```C++
```

utility

Example:

```
#include <utility>
```

cmath

Example:

```
#include <cmath>
```

string

Example:

```
#include <string>
```

string

Example:

```
#include <string>
```

ios

Example:

```
#include <ios>
```

cmath

Example:

```
#include <cmath>
```

unordered_map

Example:

```
#include <unordered_map>
```

unordered_map

Example:

```
#include <unordered_map>
```

unordered_set

Example:

```
#include <unordered_set>
```

unordered_set

Example:

```
#include <unordered_set>
```

What header is `std::ws` declared in?

C++

Given `std::vector<> v`, what C++ function checks if it's origanized as a **heap**?

C++

What is the runtime complexity of `std::is_heap`?

C++

What does the code below do?

```
std::vector<int> v{...};
std::make_heap(v.begin(), b.end());
```

C++

What is the runtime complexity of `std::make_heap`?

C++

What type of binary heap does `std::make_heap` make by default?

A *min* heap or a *max* heap?

C++

Given a `std::vector<>` as a heap, how do you remove the top element of the heap?

C++

What does `std::pop_heap` do?

C++

What is the runtime complexity of `std::pop_heap`?

C++

Given a `std::vector<>` as a heap, what C++ function adds an additional element?

C++

std::is_heap

Example:

```
std::vector<int> v = {4, 8, 9};
bool b = std::is_heap(
    v.begin(), v.end());
```

Converts the array v to be ordered as a heap data structure.

A max heap.

Move the first element of the heap to the end, while maintaining the heap.

Push to the end of the vector, and then call...

std::push_heap()

Example:

```
std::vector<int> v = {4, 8, 9};
v.push_back(1);
std::push_heap(v.begin(), v.end());
```

`istream`

Example:

```
#include <istream>
```

$O(n)$

AKA: Linear

O(N)
(linear)

(Specifically 3×N)

Call...

`std::pop_heap()`

And then remove the back element.

Example:

```
std::vector<int> v = {4, 8, 9};
std::pop_heap(v.begin(), b.end());
v.pop_back();
```

$O(\log n)$

Specifically 2×log n
AKA: Logarithmic

What is the runtime complexity of `std::push_heap`?

C++

What does `std::sort_heap()` do?

C++

What is the runtime complexity of `std::sort_heap()`?

C++

What is the **worst** case **time** complexity of **introsort**?

C++

What is the **worst** case **time** complexity of **`std::sort`**?

C++

What algorithm does `std::sort` use?

C++

What sub-sorting algorithms does introsort use?

C++

Describe what a `std::multimap<>` container does.

C++

What C++ container is similar to a `std::map`, but can hold multiple of the same value?

C++

Describe what a `std::multiset<>` container does.

C++

Sorts a (vector) container that is organized as a heap, in an optimized way.

ANSWER

$O(\log n)$

ANSWER

$O(n \log n)$

ANSWER

AKA: linearithmic

$O(n \log n)$

ANSWER

(Specifically $2n \log n$)
(Worst case)

Introsort.

ANSWER

$O(n \log n)$

ANSWER

`std::sort` uses an introsort algorithm

It's a container that holds items similar to a `std::map`, but can hold multiple of the same value.

ANSWER

First quick sort,
then heap sort,
then insertion sort.

ANSWER

It's a container that holds items similar to a `std::set`, but can hold multiple items of the same value.

ANSWER

`std::multimap<>`

ANSWER

What C++ container is **similar** to a `std::set`, but can hold **multiple** items of the **same** value?

C++

When sorting, is a **greater** than predicate ascending or decending?

```C++
std::vector<int> x = {5,1,3,2,4};
std::sort(x.begin(), x.end(),
    std::greater<int>());
```

When sorting, is a **less** than predicate ascending or descending?

```C++
std::vector<int> x = {5,1,3,2,4};
std::sort(x.begin(), x.end(),
    std::less<int>());
```

For C++,

For `std::priority_queue::push`, what is the **time** complexity?

C++

What implements a `std::priority_queue` behind the scenes?

Example:

```C++
std::priorityqueue pq;
```

For C++,

what std **container sorts** and **queues** a collection of elements?

C++

What are the valid parameters for `std::setbase`?

Example:

```C++
std::cout << std::setbase(?);
```

What is the string output from the sample below?

```C++
std::cout
    << std::setfill('*');
    << std::setw(12)
    << 5;
```

What does `std::showpoint` do?

Example:

```C++
std::cout << std::showpoint;
std::cout << 1.0;
```

What does `std::noshowpoint` do?

Example:

```C++
std::cout << std::noshowpoint;
std::cout << 1.0;
```

descending

`std::multiset<>`

$O(\log n)$

AKA: Logarithmic

ascending

A `std::priority_queue`

Defined in `<queue>`

A std::vector

and

standard library
heap functions

e.g., `std::push_heap`, `std::pop_heap`

`**********5`

`0`, `8`, `10` and `16`

(Any other value defaults to `0`)

Sets the stream state to show the decimal point for output **only if needed**.

Sets the stream state to **always** show the decimal point for output.

What does `std::string::`**`substr`** do?

And what are its **parameters**?

C++

What **member** of **`std::string`** returns a **substring**?

C++

For a `std::unordered_map<>`, what is the **time** complexity for **insertion**?

C++

For a `std::unordered_map<>`, what is the **time** complexity for **removal**?

C++

For a `std::unordered_map<>`, what is the **time** complexity for **searching**?

C++

For a `std::unordered_set<>`, what is the **time** complexity for **insertion**?

C++

For a `std::unordered_set<>`, what is the **time** complexity for **removal**?

C++

For a `std::unordered_set<>`, what is the **time** complexity for **searching**?

C++

`std::string::substr`

Example:

```
std::string s = "Firetruck yea!";
std::cout << s.substr(4,9);
> truck yea
```

Given a std::string,
it returns a sub-string of it.

params:
• pos (size_t) - The start of where to extract the substring.
• len (size_t) - The length of the substring.
returns:
string - the substring.

$O(1)$

(On average)

AKA: Constant

ANSWER

$O(1)$

(On average)

AKA: Constant

ANSWER

$O(1)$

(On average)

AKA: Constant

ANSWER

O(1)

(On average)

AKA: Constant

ANSWER

O(1)

(On average)

AKA: Constant

ANSWER

$O(1)$

(On average)

AKA: Constant