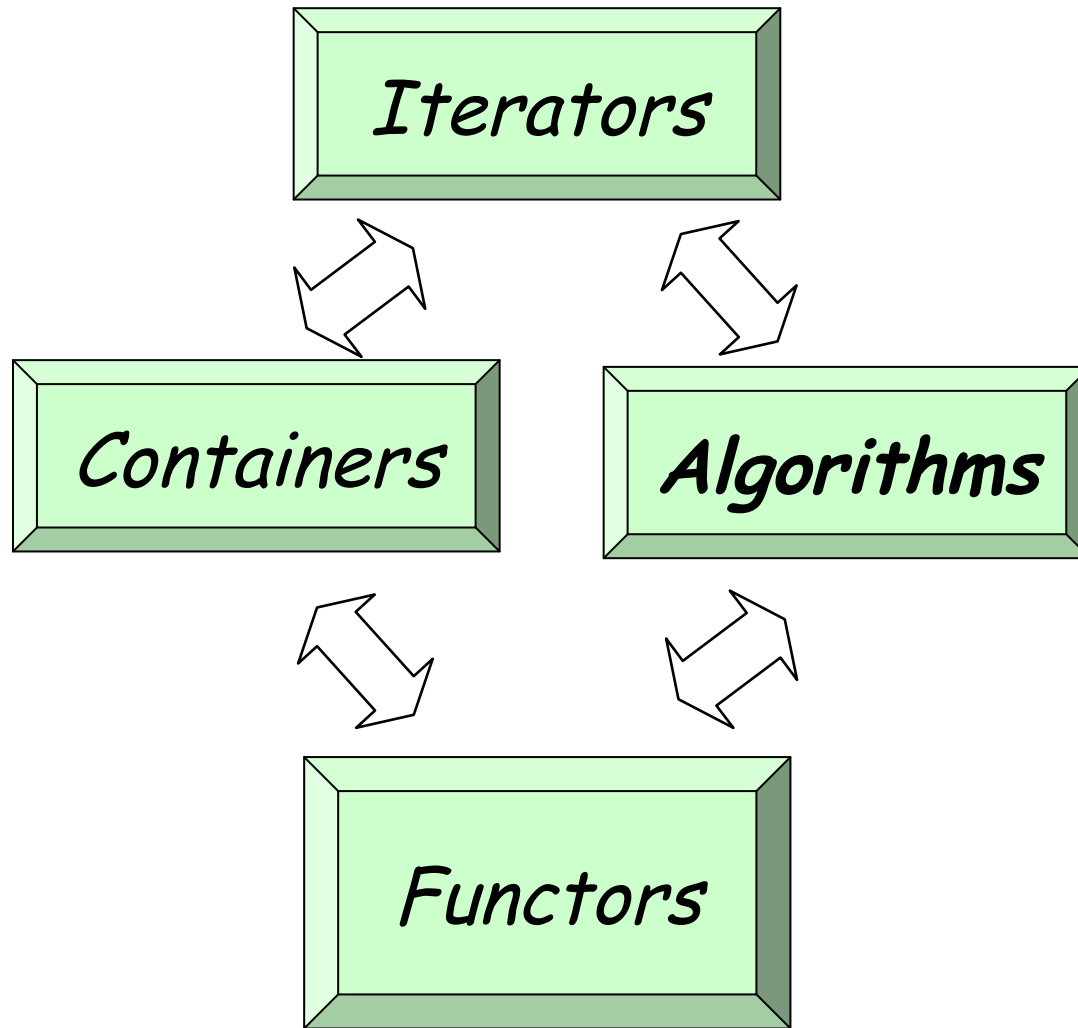


# **The Standard C++ Library – Algorithms**

*Based on:* Jonathan Boccara,  
"105 algorithms in less than an hour",  
CPPCON 2018,  
<https://youtu.be/2olsGf6JlKU>

*Summarized by:* Erel Segal-Halevi

# Main Components of STL



# *Algorithms*

- In C++17, there are 105 algorithms that perform common programming tasks.
- We need to know them all.
- *Why?* - otherwise, we will try to implement them ourselves as part of larger projects, and:
  - Make bugs (no time to test..),
  - Produce inefficient code (no time to optimize..)
- *How can we remember them?* - use a map →



PROVINCE OF  
VALUE  
QUERIES

PROVINCE OF  
PROPERTY  
QUERIES

PROVINCE OF  
2-RANGES PROPERTIES

  
GLORIOUS COUNTY OF  
ALGOS ON SETS

LONELY ISLANDS

  
LANDS OF  
QUERIES


  
PENINSULA OF  
RAW MEMORY

  
TERRITORY OF  
MOVERS

PROVINCE OF  
RESEARCH

RELATIVE VALUE  
SEARCHERS


  
ISLAND OF  
STRUCTURE  
CHANGERS

  
LAND OF  
VALUE  
MODIFIERS

SECRET RUNES

  
LANDS OF  
PERMUTATIONS

Fluent C++  
fluentcpp.com

  
THE WORLD  
OF  
C++ STL  
ALGORITHMS  
FLUENT C++

<https://youtu.be/2olsGf6JlKU?t=475>



## Queries (10)

Algorithms that calculate a value related to the collection, without modifying it.



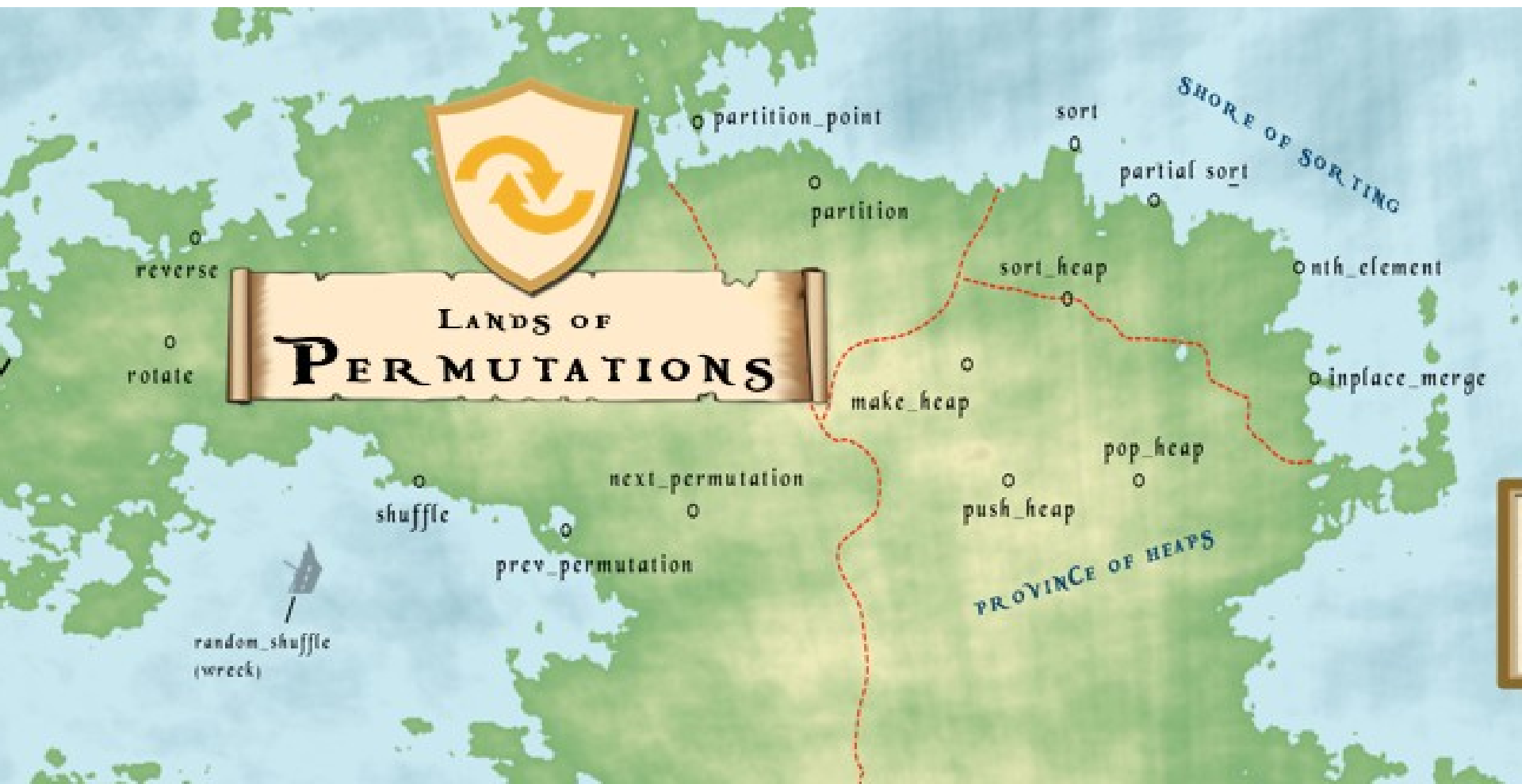
## *Set Algorithms (11)*

Operations on any sorted collection  
(including `std::set` but also sorted vectors)



## *Permutations (12)*

Algorithms that move elements around the collection, without changing their values:



## *Movers* (13)

Algorithms that copy/move things between collections.





## *Value modifiers (14)*

Modify values in a collection.



## *Runes (15)*

Words you can add to an algorithm to get a new algorithm:



## *Structure changers (16)*

STL algorithms  
work on iterators,  
so they cannot  
change the size!

*Solution:* move the  
remaining elements  
to the *beginning* of  
the collection, and  
return an iterator to  
the *new end*.



## *Others* (17)





# What now?

- Use algorithms instead of loops.
- Read the documentation: preconditions, postconditions, complexity
- Invent your own algorithms!

