

# 第七讲

*Map, Set, Lexicon*

---

薛浩

2023 年 4 月 11 日

[www.stickmind.com](http://www.stickmind.com)

- 话题 1：编程基础** 初学编程的新手，一般应该熟练使用函数和库处理字符串相关的编程任务。
- 话题 2：抽象数据类型的使用** 在尝试实现抽象数据类型之前，应该先熟练使用这些工具解决问题。
- 话题 3：递归和算法分析** 递归是一种强有力的思想，一旦掌握就可以解决很多看起来非常难的问题。
- 话题 4：类和内存管理** 使用 C++ 实现数据抽象之前，应先学习 C++ 的内存机制。
- 话题 5：常见数据结构和算法** 在熟练使用抽象数据类型解决常见问题之后，学习如何实现它们是一件很自然的事情。

## 话题 2: 抽象数据类型的使用

在尝试实现抽象数据类型之前，应该先熟练使用这些工具解决问题。

- Vecotr、Grid、Stack、Queue
- Map、Set、Lexicon

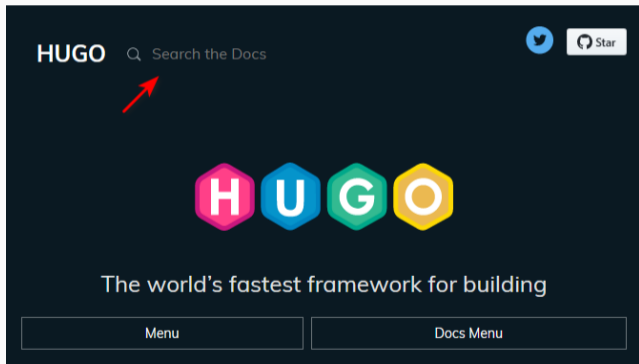


Figure 1: 搜索引擎

**如何使用计算机对实际问题建模？**

# 目录

1. 复习：容器 Container
2. 映射 Map
3. 集合 Set
4. 词典 Lexicon
5. 挑战：Word Ladder

## 复习：容器 Container

---

*As you learn more about programming, you will discover that particular data structures are so useful that they are worth studying in their own right. Moreover, it is usually far more important to know how to use those structures effectively than it is to understand their underlying representation.*

*—Eric S. Roberts, Stanford University*

**抽象数据类型**，简称为 ADT，是由基本类型不断组合而成，并由其行为定义，而不是内部表示。

Vector 一种类似于数组功能的容器，支持自动调整大小和越界检查。

Grid 针对嵌套 Vector 的一个封装，便于开发二维结构应用。

Stack 一种遵循 LIFO 顺序的容器，元素的删除和其添加顺序相反。

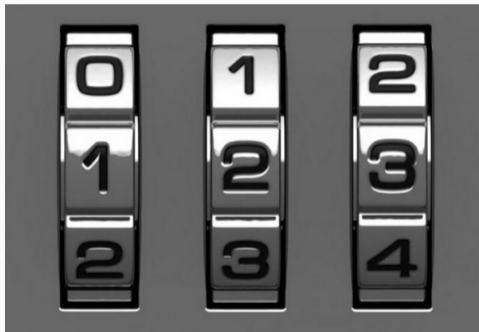
Queue 一种遵循 FIFO 顺序的容器，元素的删除和其添加顺序相同。



现实世界的问题太过于复杂而难以理解，在编程中，通常使用一个理想化**模型**来简化复杂的问题。

在不影响问题本质的前提下，构建一个合理的模型，动态地模拟其在现实世界中的行为。这样的程序称为**仿真**。

## 练习: Unlock



## 映射 Map

---

# Map 映射

Map 提供了一个**键** (key) 和**值** (value) 关联的容器。

- 键的类型必须是可比较的类型
- 值可以是基本类型，也可以是更复杂的数据结构

如果 Map 只存储变量的名字和其对应值之间的关系，那么这种映射也称为**符号表** (symbol table)。

## Map 常用方法

<b>map.size()</b> Returns the number of key/value pairs in the map.
<b>map.isEmpty()</b> Returns <code>true</code> if the map is empty.
<b>map.put(key, value)</b> <i>or</i> <b>map[key] = value;</b> Makes an association between <code>key</code> and <code>value</code> , discarding any existing one.
<b>map.get(key)</b> <i>or</i> <b>map[key]</b> Returns the most recent value associated with <code>key</code> .
<b>map.containsKey(key)</b> Returns <code>true</code> if there is a value associated with <code>key</code> .
<b>map.remove(key)</b> Removes <code>key</code> from the map along with its associated value, if any.
<b>map.clear()</b> Removes all key/value pairs from the map.

### **map[key]**

Selects the value associated with key. This syntax makes it easy to think of a map as an "associative array" indexed by the key type. If key is already present in the map, this function returns a reference to its associated value. If key is not present in the map, a new entry is created whose value is set to the default for the value type.

## 辨析：此 Map 非彼 map

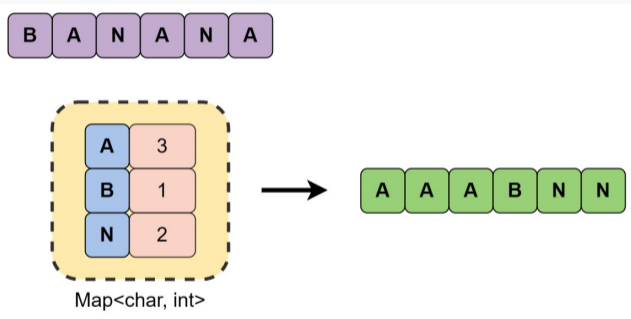
方法 `put` 将特定的键和值进行关联。如果 `key` 不存在，则插入新值；如果 `key` 已存在，则替换旧值。

```
map.put(key, value);
```

### 辨析

这个行为和标准库中的 `std::map` 不同。对于标准库中的 `map`，如果 `key` 已存在，则不作任何操作。

# Map 存储顺序





## 集合 Set

---

## Set 集合

Set 可以建模数学上的集合概念，用于表示无序且没有重复的元素集。

<b>set.size()</b> Returns the number of elements in the set.
<b>set.isEmpty()</b> Returns <code>true</code> if the set is empty.
<b>set.add(value)</b> Adds <code>value</code> to the set.
<b>set.remove(value)</b> Removes <code>value</code> from the set.
<b>set.contains(value)</b> Returns <code>true</code> if the set contains the specified value.
<b>set.clear()</b> Removes all words from the set.
<b>s1.isSubsetOf(s2)</b> Returns <code>true</code> if <code>s1</code> is a subset of <code>s2</code> .
<b>set.first()</b> Returns the first element of the set in the ordering specified by the value type.

$s_1 + s_2$	Returns the <i>union</i> of $s_1$ and $s_2$ , which consists of the elements in either or both of the original sets.
$s_1 * s_2$	Returns the <i>intersection</i> of $s_1$ and $s_2$ , which consists of the elements common to both of the original sets.
$s_1 - s_2$	Returns the <i>set difference</i> of $s_1$ and $s_2$ , which consists of the all elements in $s_1$ that are not present in $s_2$ .
$s_1 += s_2$ $s_1 -= s_2$ $s_1 *= s_2$	The $+$ , $-$ , and $*$ operators can be combined with assignment just as they can with numeric values. For $+=$ and $-=$ , the value $s_2$ can be a set, a single value, or a list of values separated by commas.

## 词典 Lexicon

---

Lexicon 是用于存储单词集合的一个定制版，目的是提升效率。

<b>lexicon.size()</b> Returns the number of words in the lexicon.
<b>lexicon.isEmpty()</b> Returns <code>true</code> if the lexicon is empty.
<b>lexicon.add(word)</b> Adds <code>word</code> to the lexicon, always in lowercase.
<b>lexicon.addWordsFromFile(filename)</b> Adds all the words in the specified file to the lexicon.
<b>lexicon.contains(word)</b> Returns <code>true</code> if the lexicon contains the specified word.
<b>lexicon.containsPrefix(prefix)</b> Returns <code>true</code> if the lexicon contains any word beginning with <code>prefix</code> .
<b>lexicon.clear()</b> Removes all words from the lexicon.

# 练习: Word Unscrambler

**WORD SCRAMBLE** **ABCyo**

## EASTER

Rearrange the letters to unscramble the words below.

 **STEBAK** **DYNCA**  
-----

**PRISGN** **GEG TUNH**  
-----

**KIHCC** **WOLFRE**   
-----

**TARCOR** **THOOLECCA**  
-----

**BLAM**  **NUNBY**  
-----









VISIT [ABCyo.com](http://ABCyo.com) FOR MORE FUN & EDUCATIONAL GAMES AND ACTIVITIES!  
© 2020 ABCyo.com, L.L.C., a division of DL Learning • All Rights Reserved


## 挑战: Word Ladder

---

# 挑战：Word Ladder

**The Word Ladder Challenge**  
How do you turn a CAT into a DOG? Have some fun with short vowels by completing this word ladder. Fill in the missing letter for each word and watch as "cat" becomes "dog" one letter at a time.

	C	A	T	
	C	A	_	
	_	A	P	
	M	_	P	
	_	O	P	
	P	O	_	
	_	O	T	
	D	O	G	

Created by  stickmind.com Copyright © 2012-2013 by stickmind.com More worksheets at [www.education.com/worksheets](http://www.education.com/worksheets)



**如何使用计算机对实际问题建模？**

问题?