

静态和动态数组

By 波波微课 & William Fiset

大纲

- 介绍数组并给出样例
 - 什么是数组Array
 - 数组的使用场景
 - 复杂度分析
 - 静态数组使用样例
- 动态数组实现细节
- 代码实现

介绍和样例

什么是静态数组

一个静态数组是一个包含 n 个元素的定长容器，其中的元素是**可以索引的**，范围从 $[0, n-1]$

Q: 可以索引是什么意思？

A: 意思是说，数组中的每一个槽位都是可以通过一个数字进行引用的。

静态数组的使用场景

- 1) 存储和访问顺序数据
- 2) 临时存储对象
- 3) 用作IO程序的缓冲区
- 4) 正向和反向查找表
- 5) 用于在函数结束时返回多个值
- 6) 用于在动态规划中缓存子问题的结果

复杂度

静态数组

动态数组

访问	$O(1)$	$O(1)$
查找	$O(n)$	$O(n)$
插入	N/A	$O(n)$
添加	N/A	$O(1)$
删除	N/A	$O(n)$

静态数组

A =	44	12	-5	17	6	0	3	9	100
	↓	↓	↓	↓	↓	↓	↓	↓	↓
	0	1	2	3	4	5	6	7	8

A中的每个元素都是通过各自的索引进行引用的。除此之外，没有其它办法可以访问数组中的元素。数组索引是基于0的，也就是说数组中的第一个元素在位置0。

静态数组

A =	44	12	-5	17	6	0	3	9	100
	↓	↓	↓	↓	↓	↓	↓	↓	↓
	0	1	2	3	4	5	6	7	8

A[0] = 44

A[1] = 12

A[4] = 6

A[7] = 9

A[9] => index out of bounds!

静态数组

A =	-1	12	-5	17	6	0	3	9	100
	↓	↓	↓	↓	↓	↓	↓	↓	↓
	0	1	2	3	4	5	6	7	8

A[0] = 44

A[0] := -1

A[1] = 12

A[4] = 6

A[7] = 9

A[9] => index out of bounds!

静态数组

A =	-1	12	-5	17	6	18	3	9	100
	↓	↓	↓	↓	↓	↓	↓	↓	↓
	0	1	2	3	4	5	6	7	8

A[0] = 44

A[0] := -1

A[1] = 12

A[5] := 18

A[4] = 6

A[7] = 9

A[9] => index out of bounds!

静态数组

A =	-1	12	-5	17	6	18	25	9	100
	↓	↓	↓	↓	↓	↓	↓	↓	↓
	0	1	2	3	4	5	6	7	8

A[0] = 44

A[0] := -1

A[1] = 12

A[5] := 18

A[4] = 6

A[6] := 25

A[7] = 9

A[9] => index out of bounds!

动态数组和操作

动态数组

动态数组的大小可以扩大或者缩小

A =

34	4
----	---

A.add(-7)

A =

34	4	-7
----	---	----

A.add(34)

A =

34	4	-7	34
----	---	----	----

A.remove(4)

A =

34	-7	34
----	----	----

动态数组

Q： 动态数组该如何实现？

A： 一种办法是使用静态数组来实现！

- 1) 创建一个具有初始容量的静态数组。
- 2) 向底层静态数组中添加元素，跟踪元素的个数。
- 3) 如果继续添加元素会超出容量，那么就创建一个具有两倍容量的新数组，并将原数组的内容拷贝到新数组当中去。

动态数组

假定我们创建一个初始容量为2的动态数组，然后开始向其中添加元素。。。

∅	∅
---	---

7	∅
---	---

7	-9
---	----

7	-9	3	∅
---	----	---	---

7	-9	3	12
---	----	---	----

7	-9	3	12	5	∅	∅	∅
---	----	---	----	---	---	---	---

7	-9	3	12	5	-6	∅	∅
---	----	---	----	---	----	---	---