

向量

## 抽象数据类型：从数组到向量

e2 - A2

阿圆眼快，把手一点说：“到了，就是这里。妈妈，你只管找号头，311，就是爸爸的号。”

贵贱长少，秩秩焉，莫不从桓公而贵敬之，是天下之大节也。

邓俊辉

deng@tsinghua.edu.cn

# 循秩访问

- ❖ C/C++语言中，数组元素与编号一一对应： $A[0], A[1], A[2], \dots, A[n-1]$
- ❖ 反之，元素各由**编号唯一指代，并可直接访问** //故亦称作线性数组 ( linear array )

$A[i]$ 的物理地址 =  $A + i \times s$ ,  $s$ 为单个元素占用的空间量



- ❖ 向量是数组的抽象与泛化，由一组元素按线性次序**封装**而成

- 各元素与 $[0, n)$ 内的**秩** ( rank ) 一一对应： `typedef int Rank; //call-by-rank`
- 操作、管理维护更加简化、统一与安全
- 元素类型可灵活**选取**，便于**定制复杂数据结构** `//Vector< PFCTree* > pfcForest;`

# 向量ADT接口

操作	功能	适用对象
<code>size()</code>	报告向量当前的规模（元素总数）	向量
<code>get(r)</code>	获取秩为r的元素	向量
<code>put(r, e)</code>	用e替换秩为r元素的数值	向量
<code>insert(r, e)</code>	e作为秩为r元素插入，原后继依次后移	向量
<code>remove(r)</code>	删除秩为r的元素，返回该元素原值	向量
<code>disordered()</code>	判断所有元素是否已按非降序排列	向量
<code>sort()</code>	调整各元素的位置，使之按非降序排列	向量
<code>find(e)</code>	查找目标元素e	向量
<code>search(e)</code>	查找e，返回不大于e且秩最大的元素	有序向量
<code>deduplicate(), uniquify()</code>	剔除重复元素	向量/有序向量
<code>traverse()</code>	遍历向量并统一处理所有元素	向量

# ADT操作实例

操作	输出	向量组成 (自左向右)	操作	输出	向量组成 (自左向右)
初始化			disordered()	3	4 3 7 4 9 6
insert(0, 9)	9	9	find(9)	4	4 3 7 4 9 6
insert(0, 4)	4 9	4 9	find(5)	-1	4 3 7 4 9 6
insert(1, 5)	4 5 9	4 5 9	sort()		3 4 4 6 7 9
put(1, 2)	4 2 9	4 2 9	disordered()	0	3 4 4 6 7 9
get(2)	9	4 2 9	search(1)	-1	3 4 4 6 7 9
insert(3, 6)		4 2 9 6	search(4)	2	3 4 4 6 7 9
insert(1, 7)		4 7 2 9 6	search(8)	4	3 4 4 6 7 9
remove(2)	2	4 7 9 6	search(9)	5	3 4 4 6 7 9
insert(1, 3)		4 3 7 9 6	search(10)	5	3 4 4 6 7 9
insert(3, 4)		4 3 7 4 9 6	uniquify()		3 4 6 7 9
size()	6	4 3 7 4 9 6	search(9)	4	3 4 6 7 9

## STL Vector

❖ `#include <iostream>`

`#include <vector>`

`using namespace std;`

❖ `vector<int> v; // an empty vector of integers`

`vector<int> s( 32, 63 ); // { 63, 63, 63, ..., 63 }, sum = 2016`

`s.insert( s.begin + 2, 2017 ); // { 63, 63, 2017, 63, ..., 63 }`

`s.erase( s.end - 30, s.end ); // { 63, 63, 2017 }`

`for ( i = 0; i < s.size(); i++ )`

`cout << s[i] << endl;`